

**GUIDA AL CORSO DI LAUREA
MAGISTRALE in INFORMATICA
(d.m. 270)
E MANIFESTO DEGLI STUDI**

Corso di Laurea Magistrale in Informatica
Dipartimento di Informatica
Università degli Studi di Torino

Via Pessinetto 12 - Torino

Anno Accademico 2015/2016



di.unito.it
DIPARTIMENTO DI INFORMATICA



UNIVERSITÀ
DEGLI STUDI
DI TORINO



Prefazione

Gentile lettore,

questa è la guida del corso di Laurea Magistrale in Informatica (secondo il DM 270, classe di laurea LM-18), il testo della guida è organizzato in due parti. La prima parte è la descrizione del corso di Laurea Magistrale in Informatica (secondo il DM 270, classe di laurea LM-18): questa descrizione costituisce il “Manifesto degli studi per la coorte 2015”. La seconda parte descrive invece gli insegnamenti offerti per il 2015/2016 (per tutte le coorti attive), elencando per ogni insegnamento i docenti di riferimento, il semestre di svolgimento e il programma di esame. La prima parte è quindi di interesse precipuamente per gli studenti che si immatricolano alla magistrale quest’anno (coorte 2015), che trovano nel Manifesto la descrizione del loro percorso biennale, mentre la seconda parte è di interesse anche per gli studenti della coorte precedente (2014) che devono seguire gli insegnamenti del secondo anno.

Ricordiamo che per gli studenti immatricolati negli anni precedenti vale il manifesto degli studi della loro coorte, pubblicato nell’anno di iscrizione e disponibile su questo stesso sito. Se l’anno di immatricolazione è precedente al 2014 possono essersi verificati dei cambiamenti nei corsi offerti, e gli studenti sono invitati a consultare gli aggiornamenti delle guide del loro anno di immatricolazione.

Di seguito alcune note riassuntive, con le principali informazioni di interesse:

1. Tutto quanto è scritto in questa guida, e nei successivi aggiornamenti, è vincolante per gli studenti della coorte 2015 e per il corso di laurea. La guida, unitamente ai regolamenti di Ateneo per tasse, piano carriera e esame di laurea, costituisce la base del patto “studenti-Università”: conoscerla ed averne compreso i contenuti è un obbligo dello studente.
2. La durata della Laurea Magistrale è di due anni accademici. Il corso di Laurea Magistrale è articolato in tre diversi curricula, al fine di permettere allo studente una maggiore personalizzazione degli studi. La scelta del curriculum avviene al momento dell’iscrizione, in quanto i percorsi si differenziano già dal primo semestre del primo anno.
3. L’ammissione al corso di Laurea Magistrale richiede (almeno) un titolo di laurea triennale.
4. L’ammissione al corso di Laurea Magistrale in Informatica viene deliberata da apposita commissione dopo la valutazione dell’adeguatezza del curriculum studiorum (tipo di studi triennali effettuati e relativi insegnamenti, nonché eventuali altre esperienze di formazione) ed un colloquio con il candidato. Il curriculum studiorum è sempre considerato adeguato per i laureati triennali in Informatica che conseguano il titolo presso l’Università degli Studi di Torino.
5. La Laurea Magistrale dà accesso, previo esame di selezione, al Dottorato di Ricerca in Informatica, della durata di tre anni.
6. La Laurea Magistrale dà accesso ai corsi di master di secondo livello, normalmente della durata di un anno. I master hanno un obiettivo maggiormente professionalizzante e vengono organizzati in collaborazione con aziende del settore e altri enti formativi, e la loro offerta varia di anno in anno.
7. L’iscrizione agli esami è obbligatoria e avviene tramite una procedura telematica, attiva sul sito web dell’Università di Torino.

8. La valutazione degli insegnamenti avviene con procedura telematica. La valutazione di un insegnamento da parte dello studente è condizione necessaria per l'iscrizione all'esame dell'insegnamento stesso.
9. Tutte le informazioni non riportate in questa guida (come i programmi dettagliati degli insegnamenti, gli orari, la localizzazione delle aule, tasse e scadenze) sono sul [sito Internet del Corso di Laurea Magistrale](#), o sul sito di Ateneo www.unito.it

Buona lettura!

Susanna Donatelli, PhD,
Professore Ordinario di Informatica,
Presidente del Corso di Laurea e Laurea Magistrale in Informatica
Tel. 011 6706711
e-mail presccs@educ.di.unito.it

INDICE

Prefazione	2
Prima parte: Manifesto degli studi	5
Obiettivi e sbocchi professionali.....	5
Obiettivi del Corso di Laurea Magistrale.....	5
Sbocchi professionali	5
Certificazione di Qualità	6
Organizzazione della didattica.....	7
Introduzione	7
Requisiti e modalità di ammissione	7
Maurizio Lucenteforte	9
Giancarlo Ruffo	9
Pietro Torasso	9
Periodi di lezione	9
Esami.....	10
Edumeter e il processo per la qualità del Corso di Laurea Magistrale.	10
Piano Carriera	10
Prova finale	11
Struttura del corso di Laurea Magistrale e curricula previsti.....	12
Indicazioni comuni a tutti i curricula	12
Struttura dell’offerta formativa rispetto al RAD della classe	13
Curriculum “Realtà Virtuale e Multimedialità”	15
Curriculum “Reti e Sistemi informatici”	17
Curriculum “Sistemi per il Trattamento dell’Informazione”	20
Dopo la magistrale: il dottorato di ricerca	24
Il Dottorato di Ricerca in Informatica presso il Dipartimento di Informatica	24
Organizzazione del Dottorato	24
Informazioni.....	24
Supporto agli studi	25
Erasmus.....	25
Biblioteca	25
Supporto on-line agli insegnamenti	26
Aule e laboratori	26
Zone studio.....	26
Supporto agli studenti Disabili.....	27
Supporto agli studenti con DSA (Disturbi Specifici di Apprendimento)	28
Indirizzi utili.....	29
Seconda Parte: Programmi e docenti degli insegnamenti per l’A.A. 15/16	30
Insegnamenti attivi nel 15/16.....	30
Insegnamenti mutuati attivi nell’A.A. 15/16	32
Insegnamenti ad anni alterni non tenuti nel 15/16:.....	34
Insegnamenti esterni all’Ateneo.	34

Prima parte: Manifesto degli studi

Obiettivi e sbocchi professionali

Obiettivi del Corso di Laurea Magistrale

È obiettivo primario della laurea magistrale in Informatica formare laureati con un'approfondita comprensione e conoscenza dell'Informatica nei suoi vari e articolati aspetti. Tale obiettivo è perseguito attraverso l'articolazione del corso di laurea magistrale in curricula (indirizzi), che, pur condividendo diversi insegnamenti, permettono una significativa caratterizzazione delle competenze acquisite, in modo da rispondere nel modo più adeguato agli interessi degli studenti e agli sbocchi professionali attualmente presenti nel mercato. Così, mentre il corso di laurea triennale fornisce conoscenze di base accanto a elementi di formazione professionalizzante, al fine di consentire un inserimento diretto nel mondo del lavoro, il corso di laurea magistrale permette di ottenere una più spiccata specializzazione in importanti settori dell'informatica di base e di proiettarsi maggiormente verso aspetti interdisciplinari, ma soprattutto di acquisire maggiori capacità di comprendere, analizzare in modo critico e/o sviluppare nuove tecnologie, aspetto estremamente importante della propria personale formazione in un ambito in continua evoluzione come l'informatica.

È obiettivo altrettanto importante per la nostra laurea magistrale che essa fornisca una base di conoscenze adeguata a quegli studenti che, indipendentemente dal curriculum scelto, intendono poi perfezionare la propria formazione con studi di terzo livello, quali il Dottorato di Ricerca. In particolare lo svolgimento della tesi di laurea magistrale potrà comportare la partecipazione degli studenti a progetti di ricerca coordinati da docenti del corso di laurea magistrale, spesso in collaborazione con aziende e enti di ricerca. La partecipazione attiva a progetti di ricerca costituirà per gli studenti un'occasione di autovalutazione della propria propensione per l'attività di ricerca avanzata, e consentirà loro di scegliere con piena consapevolezza se continuare gli studi col Dottorato di Ricerca o con corsi di Master.

Allo stato attuale, tenendo conto sia dei diversi ruoli che l'informatica gioca nei servizi e nei processi produttivi, che delle competenze scientifiche e didattiche presenti nell'Area Informatica dell'Ateneo, sono stati individuati i seguenti temi di approfondimento: linguaggi e metodi di sviluppo del software, realtà virtuale e multimedialità, reti e sistemi informatici, sistemi per il trattamento dell'informazione. A partire da questi temi sono stati attivati tre curricula: Realtà Virtuale e Multimedialità (RVM), Reti e Sistemi Informatici (RSI), Sistemi per il Trattamento dell'Informazione (STI)

Sbocchi professionali

I laureati magistrali in Informatica svolgeranno attività professionale nella progettazione, organizzazione, gestione e manutenzione di sistemi informatici complessi e/o innovativi, con specifico riguardo ai requisiti di qualità, affidabilità, efficienza e sicurezza. I laureati saranno in grado, nel breve e medio periodo, di coordinare, dirigere e controllare progetti di definizione, di sviluppo o di acquisizione, di messa in campo e di gestione di sistemi informatici integrati per il trattamento dei dati e dei processi aziendali, di ideare e gestire servizi di rete di aziende e

strutture pubbliche e private anche di grandi dimensioni, di fornire supporto alle scelte della direzione in materia di automazione e di informatizzazione dei processi.

I laureati magistrali potranno trovare impiego in imprese che sviluppano sistemi informatici, ma anche in enti di ricerca e società high-tech che sviluppano nuove metodologie e sistemi informatici innovativi, ovvero che utilizzano soluzioni informatiche avanzate per innovare i loro prodotti anche in ambiti interdisciplinari che spaziano dalle applicazioni finanziarie, bancarie ed assicurative, alla tutela dell'ambiente, ai media, allo spettacolo ed all'intrattenimento ed infine in imprese ed organizzazioni pubbliche e private che fanno uso di reti e sistemi informatici complessi per gestire i propri dati e processi. Le competenze acquisite durante il corso di laurea magistrale consentiranno al laureato anche di avviare attività professionali autonome o di proseguire il percorso di studi con formazione di terzo livello. La laurea magistrale dà inoltre accesso ai percorsi di abilitazione all'insegnamento nelle scuole, secondo la normativa vigente.

Il corso prepara alle seguenti **professioni ISTAT**: Direttori e dirigenti del dipartimento servizi informatici, Imprenditori e responsabili di piccole aziende nei servizi alle imprese e alle persone, Analisti e progettisti di software, Analisti di sistema, Analisti e progettisti di applicazioni web, Specialisti in reti e comunicazioni informatiche, Analisti e progettisti di basi dati, Amministratori di sistemi, Specialisti in sicurezza informatica, Ricercatori e tecnici laureati nelle scienze matematiche e dell'informazione

Il D.P.R. 328/2001 ha istituito nell'Albo professionale degli Ingegneri le Sezioni A e B, , in relazione al diverso grado di capacità e competenza acquisita mediante il percorso universitario (magistrale e triennale rispettivamente). Ciascuna sezione è ripartita nei seguenti Settori: Civile e Ambientale, Industriale e dell'Informazione. La Laurea magistrale in Informatica è una delle lauree previste per l'ammissione all'esame di stato per la professione di **Ingegnere dell'Informazione, Sez. A**.

Certificazione di Qualità

Il GRIN, che è l'organizzazione nazionale dei ricercatori di Informatica, ha istituito, dall'anno 2004, un certificato che attesta la qualità dell'organizzazione didattica dei Corsi di Laurea e Laurea Magistrale in Informatica italiani. Il nostro Corso di Laurea Magistrale ha già ricevuto tale certificato nazionale di qualità negli anni scorsi ed è in corso la procedura di acquisizione per il 2015-2016.

Informazioni più dettagliate sui criteri e gli obiettivi di questa certificazione si possono ottenere direttamente sul [sito GRIN](#).

Organizzazione della didattica

Introduzione

Il *Corso di Laurea Magistrale in Informatica* è un percorso biennale che porta al conseguimento del titolo di Dottore Magistrale in Informatica; l'iscrizione è riservata agli studenti già in possesso di un titolo di laurea triennale.

Il percorso di studi di 5 anni, laurea seguita da magistrale (anche noto come percorso "3+2"), può essere completato dal *Dottorato di Ricerca*, la cui finalità è di preparare i giovani laureati alla ricerca scientifica, o da *Master di secondo livello*. Il Dottorato di Ricerca ha cadenza annuale e durata triennale, mentre l'offerta dei Master, normalmente concordata con le aziende, anche in risposta a bandi regionali o europei, non ha una cadenza, una data di inizio e un ambito tematico ricorrente, e quindi le informazioni non sono riportate in questa guida.

Un concetto fondamentale per comprendere l'organizzazione dei corsi di laurea è quello di *Credito Formativo Universitario* (CFU). Ricordiamo che le norme di legge prevedono che ogni CFU equivalga a 25 ore di lavoro per lo studente. Le 25 ore comprendono sia le lezioni (e le esercitazioni), sia lo studio individuale. Un Corso di Laurea Magistrale consta di 120 CFU. Per il Corso di Laurea Magistrale in Informatica si è stabilito che ogni CFU sia equivalente a 10 ore di lezione/esercitazione in aula o in laboratorio + 15 ore di studio individuale.

Ad esempio, un insegnamento di 6 crediti corrisponderà a 60 ore di lezioni ed esercitazioni, e si assume che allo studente (che abbia conoscenza dei requisiti di base) siano richieste ulteriori 90 ore per lo studio, i ripassi, la preparazione dell'esame, ecc. In tutte le tabelle che seguono, la "durata" degli insegnamenti è espressa in CFU. È importante ricordare che questi conteggi fanno riferimento ad uno studente "tipo" che abbia acquisito nella laurea triennale una solida preparazione di base nei vari campi dell'informatica, unitamente ad adeguate capacità di progettazione e programmazione. In questa guida, i termini *corso*, *insegnamento* e *unità didattica* sono usati in modo equivalente.

Requisiti e modalità di ammissione

Gli studenti e le studentesse interessati ad iscriversi al Corso di Laurea Magistrale in Informatica devono seguire le modalità previste per tutti i corsi di laurea dell'Ateneo, in ottemperanza alle normative ministeriali, che per l'anno 2015/2016 sono così articolate:

1. **Domanda di ammissione preliminare per la verifica dei requisiti curriculari**
2. **Colloquio di ammissione, volto ad accertare la personale preparazione.**
3. **Conferma iscrizione ON-LINE**

1. Domanda di ammissione preliminare per la verifica dei requisiti curriculari

Questo primo passo è una fase istruttoria che prevede la registrazione dello studente / della studentessa e la verifica dei suoi requisiti curriculari ai fini dell'iscrizione alla magistrale.

Gli studenti e le studentesse devono come prima cosa identificarsi (fare login) sul portale di Ateneo con le proprie credenziali personali. Chi non ha ancora delle credenziali valide per il portale di Ateneo deve prima registrarsi (www.unito.it). Una volta avvenuta l'identificazione, lo studente / la studentessa chiede la verifica dei requisiti curriculari scegliendo l'opzione "iscrizione al test di valutazione". Per Informatica non si tratta di un test in senso stretto: la studentessa / lo studente fa semplicemente l'upload dei documenti richiesti, indicando anche quali siano le sue preferenze di date per il successivo colloquio.

La Commissione di Ammissione del Corso di Laurea Magistrale di Informatica verifica il possesso dei requisiti curriculari, comunica via mail alla studentessa o allo studente l'esito della verifica e, qualora questo sia positivo, invia la convocazione al colloquio per l'accertamento della preparazione personale.

Si noti che **la verifica dei requisiti curriculari è obbligatoria per tutti**, anche per i laureati e le laureate della classe L31 (Informatica) del nostro Ateneo.

La verifica dei requisiti curriculari è garantita dare esito positivo per chi abbia conseguito la laurea (triennale) in Informatica presso l'Università di Torino. Sulla base dell'esperienza maturata in questi anni l'adeguatezza del curriculum è riconosciuta nella grande maggioranza dei casi agli studenti che hanno conseguito una laurea (triennale) in informatica (Classe L-31 nell'ordinamento del D.M. 270/04 e Classe 26 nell'ordinamento del D.M. 509/99) presso un altro ateneo italiano, in particolare qualora si tratti di laurea con certificazione "bollino blu" GRIN-AICA.

Per i laureati in altre classi di laurea, la valutazione della adeguatezza del curriculum terrà in particolare conto la presenza di un adeguato numero di CFU ottenuti nei settori scientifico disciplinari (SSD) S.S.D INF/01, informatica, o ING-INF/05, ingegneria informatica, oltre che nel settore matematico.

Gli studenti e le studentesse in possesso di titolo di studi conseguito all'estero dovranno rivolgersi in tempo utile all'Ufficio Studenti Stranieri e Traduzioni.

2. Colloquio di ammissione:

Le date già fissate per i colloqui di ammissione preliminare sono le seguenti:

- martedì 8 settembre 2015 ore 14.00 presso la sala seminari del Dipartimento di Informatica
- martedì 29 settembre 2015 ore 14.00 presso la sala seminari del Dipartimento di Informatica

I candidati saranno comunque avvisati personalmente della data in cui si devono presentare. Ulteriori date verranno comunicate in seguito sul sito web del corso di studi. L'ultima data utile per il sostenimento del colloquio di ammissione sarà intorno alla metà di dicembre 2015, nella settimana che precede la scadenza delle iscrizioni.

Durante il colloquio di ammissione verranno accertate le conoscenze sui seguenti argomenti:

- o programmazione secondo i principali paradigmi e linguaggi
- o architetture hardware e software
- o reti di elaboratori e sicurezza
- o gestione di dati e conoscenza
- o sistemi operativi
- o basi di dati e sistemi informativi
- o algoritmi
- o linguaggi formali, calcolabilità e complessità
- o matematica discreta e del continuo

La commissione ammissioni comunica immediatamente al candidato o alla candidata l'avvenuto superamento del colloquio di ammissione.

Si ricorda che il corso di studi non è a numero chiuso e pertanto il colloquio non è volto a stilare classifiche, ma ad assicurare che tutti gli iscritti alla magistrale abbiano le competenze adeguate a seguire con profitto gli studi magistrali. A tal fine è possibile che la commissione ammissioni vincoli in parte il piano carriera del candidato. La violazione di tali vincoli al

momento del caricamento del piano carriera da parte dello studente rende il piano non valido ai fini del conseguimento del titolo.

3. Conferma iscrizione ON-LINE:

Lo studente / la studentessa, dopo il conseguimento della Laurea e il superamento del colloquio di ammissione, potranno confermare l'iscrizione on-line sul portale di Ateneo **dal 1° settembre 2015 al 23 dicembre 2015**

Gli studenti che conseguano la laurea dopo il termine delle iscrizioni magistrali, non possono iscriversi alla magistrale per il 15/16, ma possono comunque rivolgersi alla Commissione di Ammissioni della Laurea Magistrale all'indirizzo ammissione-lm18@educ.di.unito.it per una valutazione dell'adeguatezza del curriculum. Qualora il curriculum venga considerato pienamente adeguato, lo studente può acquisire crediti di insegnamenti concordati con la Commissione di Ammissione, tramite il meccanismo dei corsi singoli e fino ad un massimo di 30 CFU. Tali crediti potranno essere riconosciuti al momento dell'iscrizione ed entrare nel novero dei 120 cfu necessari al conseguimento del titolo. Il suddetto meccanismo dei "corsi singoli", con la stessa limitazione dei 30 CFU annuali, potrà essere utilizzato anche da coloro i quali necessitino di integrazioni per il raggiungimento dell'adeguatezza del curriculum.

I referenti di indirizzo per Corso di Laurea Magistrale in Informatica, sono:

Indirizzo	Docente	Telefono	mail
Realtà virtuale e Multimedialità	<i>Maurizio Lucenteforte</i>	011 – 670 68 30	lucente@di.unito.it
Reti e Sistemi Informatici	<i>Giancarlo Ruffo</i>	011 – 670 67 71	ruffo@di.unito.it
Sistemi per il Trattamento dell'Informazione	<i>Pietro Torasso</i>	011 – 670 67 09	torasso@di.unito.it

Periodi di lezione

L'Anno Accademico del Corso di Laurea Magistrale in Informatica è suddiviso in due periodi didattici di 19 settimane ciascuno (*semestri*), di cui 12 settimane di lezione ed almeno 5 settimane per permettere agli studenti di sostenere gli esami. Per l'Anno Accademico 2015/2016 le date di inizio e di fine del periodo di lezioni sono le seguenti:

<p>Primo semestre: 28 settembre 2015 – 15 gennaio 2016 Sessione invernale di esami: 18 gennaio 2016 – 19 febbraio 2016 Secondo semestre: 22 febbraio 2016 - 01 giugno 2016 Sessione estiva di esami: 06 giugno 2016 – 29 luglio 2016 Sessione autunnale di esami: 01 settembre 2016 – inizio lezioni a.a. 2016-2017</p>

Gli orari e le aule degli insegnamenti sono disponibili all'indirizzo <http://di.unito.it/orarilezioni>. Si ricorda che la frequenza ai corsi non è obbligatoria, ma è comunque fortemente raccomandata, in particolare per gli insegnamenti con Laboratorio.

Esami

Per ogni unità didattica oggetto d'esame, sono previsti cinque appelli d'esame all'anno. Le date e gli appelli degli esami sono disponibili a partire da fine settembre all'indirizzo <http://di.unito.it/appelli>. Si ricorda che per poter sostenere un esame **lo studente deve obbligatoriamente iscriversi** allo stesso dal proprio sito della bacheca degli esami, disponibile sulla pagina personale dello studente sul sito di Ateneo. Le modalità di esame variano da insegnamento ad insegnamento, ma rimangono invariate per tutti e cinque gli appelli dell'anno

È possibile iscriversi all'esame di un insegnamento solo se

- si è in regola con il pagamento delle tasse e
- l'insegnamento fa parte del proprio piano carriera per l'anno in corso (o precedenti) e
- si è effettuata la valutazione dell'insegnamento stesso (procedura web Edumeter per la raccolta delle opinioni degli studenti sugli insegnamenti, si veda il punto seguente)

Sono previste tre sessioni di laurea all'anno, normalmente nei mesi di Luglio, Ottobre e Marzo/Aprile. Le date per le sessioni di laurea della coorte 2015/2016 saranno rese note entro l'inizio del terzo anno accademico della coorte.

Edumeter e il processo per la qualità del Corso di Laurea Magistrale.

Con il 2015/2016 diventa obbligatorio per tutti gli studenti (e per tutti i docenti) la valutazione della didattica. Per poter sostenere l'esame di un insegnamento lo studente deve aver *obbligatoriamente valutato* l'insegnamento stesso. La valutazione riguarda aspetti legati all'interesse suscitato nello studente dall'insegnamento, la chiarezza espositiva del docente, la regolarità nello svolgimento delle lezioni, l'adeguatezza delle strutture (aule o laboratori) in cui il corso si è tenuto e molto altro. Tale procedura è obbligatoria anche per gli studenti non frequentanti (che avranno un diverso set di domande) e per i docenti (che potranno esprimere un parere sulle strutture e sulla preparazione e interesse degli studenti). Tali valutazioni sono raccolte in forma anonima e costituiscono la base per le attività di monitoraggio della didattica erogata dal nostro corso di studio. In particolare ogni anno, in autunno, il corso di studio effettua un'attività di riesame annuale delle attività di insegnamento: si analizzano gli eventuali punti di forza e le eventuali criticità, si identificano contromisure e le si pianificano nel tempo, inoltre si verifica che contromisure precedentemente identificate siano state correttamente implementate o siano in fase di svolgimento.

Piano Carriera

Gli studenti sono tenuti a presentare il piano carriera (ex carico didattico) seguendo le istruzioni della pagina: <http://di.unito.it/pianocarriera>: il periodo per effettuare il Piano Carriera sarà dal mese di ottobre 2015 al 29 **gennaio 2016**.

Lo studente deve presentare il proprio piano carriera nei tempi descritti nella sezione "date e scadenze". Lo studente dovrà selezionare tutti gli insegnamenti del piano carriera così come stabiliti al momento dell'ammissione. Ulteriori variazioni andranno **obbligatoriamente concordate preventivamente** con la stessa commissione ammissione. Si fa notare che il sistema informatico di supporto al piano carriera non prevede alcuna funzionalità di controllo di aderenza del piano carriera del singolo con quanto concordato con la commissione ammissione: è responsabilità dello studente inserire il piano concordato in fase di ammissione. Eventuali deviazioni dal piano concordato verranno rilevate solo al momento della iscrizione all'esame di

laurea, momento in cui viene effettuato un controllo complessivo della carriera dello studente: piani non conformi a quanto concordato porteranno alla mancata iscrizione all'esame di laurea. La compilazione del Piano Carriera è condizione necessaria per poter sostenere gli esami.

Prova finale

La prova finale consiste nella preparazione e discussione di una tesi su un argomento specifico preventivamente concordato con un relatore interno che supervisionerà l'attività nelle sue diverse fasi. La discussione della tesi avviene alla presenza di una commissione nominata dalle strutture didattiche. L'attività svolta nell'ambito della tesi dovrà vertere su una delle tematiche caratterizzanti la laurea magistrale e potrà essere effettuata sia all'interno delle strutture universitarie, sia presso centri di ricerca, aziende o enti esterni, secondo le modalità stabilite dal Corso di Laurea Magistrale.

Struttura del corso di Laurea Magistrale e curricula previsti

Nel seguito viene descritta la struttura del corso di laurea magistrale, comune ai tre curricula, e come tale struttura sia declinata negli insegnamenti per i tre curricula attivati per l'A.A. 2015/2016. Vengono come prima cosa delineati alcuni aspetti comuni a tutti i curricula, e vengono poi riportate le tabelle ministeriali (RAD) relative al piano dell'offerta formativa per la classe LM-18 e la corrispondente implementazione dell'offerta effettuata dall'Università di Torino, e approvata dal Ministero. Vengono infine descritti i tre curricula, con l'elenco degli associati insegnamenti e la loro distinzione in caratterizzanti, affini e liberi, e la loro natura obbligatoria o meno.

Indicazioni comuni a tutti i curricula

1. Oltre alle possibilità di scelta fra insegnamenti caratterizzanti e affini, ogni curriculum prevede almeno 12 CFU (max 21) a scelta libera dello studente che possono essere utilizzati per approfondire la conoscenza su tematiche specifiche del curriculum oppure per esplorare altre tematiche che sono trattate in altri curricula (o altri corsi di laurea magistrale). Nell'inserire insegnamenti tra i crediti liberi gli studenti devono porgere particolare attenzione ad eventuali propedeuticità specifiche per quell'insegnamento e rispettare i punti 2, 3 e 4 seguenti. Lo studente interessato può inserire sino a 21 crediti liberi, arrivando ad un totale di 129 cfu per l'intero percorso formativo (9 in più dei 120 necessari per laurearsi).
2. Lo studente non può inserire nel Piano di Studi insegnamenti che hanno argomenti significativamente sovrapposti ad argomenti di insegnamenti già acquisiti nel percorso triennale o in altri insegnamenti della magistrale.
3. Durante il colloquio di ammissione viene verificato nel dettaglio il percorso di I Livello e la sua adeguatezza: è possibile che la Commissione ammissioni subordini l'approvazione del piano di studio complessivo all'inserimento, come crediti liberi, di insegnamenti che coprono tematiche affrontate solo in modo marginale nel curriculum di I Livello: per queste situazioni si utilizzeranno prevalentemente gli insegnamenti mutuati dal percorso di I Livello in Informatica.
4. Gli insegnamenti etichettati come "... – Parte A" si riferiscono a moduli da 6 cfu che sono mutuati dai rispettivi insegnamenti da 9 cfu: nel piano di studi non è ovviamente possibile averli entrambi.
5. Alcuni insegnamenti sono attivati ad anni alterni, questo per permettere allo studente una più ampia scelta di argomenti a parità di docenti impegnati: nelle tabelle del piano carriera dei tre curricula è indicato l'anno di attivazione. I cinque appelli di esame vengono assicurati anche per gli anni in cui l'insegnamento non si tiene.
6. I programmi degli insegnamenti e i semestri di svolgimento li potete trovare nella seconda parte della guida, quella relativa all'offerta per il 15/16. Per completezza sono elencati, separatamente, anche gli insegnamenti non attivi nell'anno, per permettere allo studente di formulare correttamente il proprio piano carriera.
7. Nelle tabelle dei curricula vi sono due campi, uno con l'anno di corso previsto, e l'altro con l'anno di corso consigliato. L'anno di corso previsto è un'indicazione prescrittiva (lo studente del primo anno non potrà iscriversi ad un esame del secondo anno), mentre l'anno di corso consigliato è come dice il nome, un suggerimento. Allo scopo di mantenere un'ampia offerta didattica e di ridurre, per quanto possibile, la presenza di sovrapposizioni di orario, si consiglia di seguire gli insegnamenti nell'anno di corso consigliato.

Struttura dell'offerta formativa rispetto al RAD della classe

La classe LM-18 prevede una forte componente informatica (INF/01) o di Ingegneria informatica e automatica (ING-INF/05), integrati da un certo numero di insegnamenti nell'area delle competenze affini ed integrative. Questa struttura è comune a tutti i corsi di laurea magistrale della classe LM-18, qualunque sia la sede. Il Corso di Laurea Magistrale di Informatica di Torino ha scelto una struttura dei piani carriera che prevede 69 cfu di insegnamenti INF/01, 12 cfu di insegnamenti per competenze affini ed integrative in ambito matematico, fisico ed economico, (almeno) 12 cfu a scelta libera dello studente, 3 cfu di altre attività (Lingua Inglese II), 24 cfu per il lavoro di tesi e la sua discussione in sede di esame di laurea. Questa struttura è comune a tutti i curricula del Corso di Laurea Magistrale, che si differenziano invece per la scelta degli esami obbligatori e consigliati per il piano carriera dello studente.

Attività caratterizzanti

ambito: Discipline Informatiche		CFU 69
C11	<u>gruppo C11</u> INF/01 Informatica	-
Totale Attività Caratterizzanti		69

Attività affini

ambito: Attività formative affini o integrative		CFU 12
A11	<u>gruppo A11</u> FIS/01 - Fisica sperimentale MAT/09 - Ricerca operativa MAT/08 - Analisi numerica MAT/06 - Probabilità e statistica matematica MAT/05 - Analisi matematica MAT/01 - Logica matematica	6 - 12
A12	<u>gruppo A12</u> SECS-P/08 - Economia e gestione delle imprese	0 - 6
Totale Attività Affini		12

Altre attività

	CFU
A scelta dello studente	12
Per la prova finale	24
Altre conoscenze utili per l'inserimento nel mondo del lavoro	3
Totale Altre Attività	39

TOTALE CREDITI**120**

Curriculum “Realtà Virtuale e Multimedialità”

La realtà virtuale è la tecnologia che permette di simulare mondi tratti dalla realtà oppure dettati dalla fantasia; in questi ultimi non necessariamente devono essere rispettate le leggi naturali alle quali siamo abituati. Per ottenere simulazioni con livelli di sofisticazione diversificati, dettati dalle esigenze realizzative, occorre sfruttare i dispositivi e le metodologie che scaturiscono dalla ricerca in questo ambito, potendo così sia dar vita ai diversi aspetti della conoscenza e comunicazione, fra i quali i più usuali sono quelli testuali, grafici, visivi e sonori, sia fornire all'utente la sensazione di essere immerso negli ambienti di sintesi, permettendogli inoltre di poter interagire con quanto in essi contenuto. Ogni azione avviene in tempo reale e richiede pertanto un'adeguata scelta di metodologie tecnologiche e algoritmiche, nonché un'oculata fusione delle stesse. La multimedialità è invece, in forma metaforica, la parente povera della realtà virtuale, ma nonostante ciò gioca un ruolo importante nella comunicazione di massa. Basti infatti pensare a come l'informazione sia sempre più resa fruibile attraverso un uso massiccio di sistemi informatici che ne permettono la rappresentazione in forme diversificate, fra loro collegate. Si tratta quindi di tecnologie hardware e software che organizzano, elaborano e veicolano i vari aspetti dell'informazione nella sua più ampia accezione.

L'obiettivo primario di questo curriculum è pertanto quello di formare professionisti in grado di analizzare e sviluppare sistemi e tecnologie nell'ambito della creazione e dell'elaborazione dell'informazione multimodale nella sua evoluzione spazio-temporale, nonché in quello della simulazione. Per raggiungere questo scopo sono state operate opportune scelte dei temi da sviluppare, ritenuti fondamentali per la cultura di un laureando che voglia affrontare il mondo del lavoro o quello della ricerca. Gli insegnamenti forniscono pertanto agli studenti gli strumenti atti a rappresentare, sintetizzare e rendere fruibile l'informazione in applicazioni che riguardino per esempio l'industria, il mondo dello spettacolo, lo sport, Internet, l'arte, la medicina. Questo curriculum, oltre ad amplificare le conoscenze informatiche di base, intende fornire un bagaglio di esperienze interdisciplinari in modo tale da indurre nello specialista una visione ampia del necessario interscambio di competenze che è fondamentale per la creazione di sistemi complessi.

La figura professionale collegata a questo curriculum di laurea magistrale è quella di esperto nella veste di sviluppatore di metodologie per la simulazione del reale o del virtuale oppure in quella di costruttore di applicazioni. Alcuni esempi di impiego sono:

- modellatore di ambienti: occorre adattare le leggi della modellazione geometrica dello spazio e degli oggetti, nonché le trasformazioni degli stessi, tenendo in considerazione specifiche necessità applicative;
- creatore di effetti di foto e fono realismo: occorre riprodurre fenomeni luminosi o sonori sulla base di condizioni da rispettare e dei risultati attesi;
- simulatore di effetti: si devono applicare leggi fisico-matematiche avanzate a sistemi e movimenti complessi;
- ottimizzatore di sistemi per l'immersione in ambienti virtuali;
- ideatore di esperienze virtuali in rete, con particolare riferimento alla cooperazione;
- analista di metodologie per la classificazione di oggetti multimodali;
- progettista di sistemi di memorizzazione e recupero di informazione multimodale.

I laureati potranno trovare impiego presso enti di ricerca, società high-tech di sviluppo, enti di tutela degli ambienti, enti per lo spettacolo e l'intrattenimento.

Articolazione del curriculum

Nel curriculum Realtà Virtuale e Multimedialità è **fondamentale l'inserimento dell'insegnamento di Metodi Numerici**, anche se questo compare in un gruppo di scelta, a meno che non siano state acquisite competenze equivalenti durante il percorso di studi triennale.

Il piano di studio consigliato è riportato nella seguente tabella, in cui “Codice” e “Insegnamento” sono il codice di Ateneo e il titolo dell'insegnamento, “SSD” è il [settore scientifico disciplinare](#) della insegnamento, “CFU” sono i crediti formativi associati all'insegnamento, “Anno” è l'anno del piano carriera in cui è previsto l'insegnamento. Qualora l'anno previsto sia “1 o 2”, quindi a scelta dello studente, la colonna “Anno consigliato” indica l'anno utilizzato per la preparazione dell'orario del corso.

Curriculum “Realtà Virtuale e Multimedialità” – Piano coorte 2015

Codice	Insegnamento	tip	area	SSD	CFU	anno	suggerito	Sem
CURRICULUM Realtà Virtuale e Multimedialità, percorso 102								
Formazione caratterizzante:								
mfn0972	Elaborazione di Immagini e Visione Artificiale	B	C11, attività caratterizzante - discipline informatiche	inf/01	9	1	1	1 sem
mfn0973	Modellazione Grafica	B	C11, attività caratterizzante - discipline informatiche	inf/01	9	1	1	2 sem
mfn0978	Sistemi di Realtà Virtuale	B	C11, attività caratterizzante - discipline informatiche	inf/01	9	1 o 2	1 o 2	1 sem
mfn0961	Teoria dell'Informazione	B	C11, attività caratterizzante - discipline informatiche	inf/01	9	1	1	2 sem
Tre Insegnamenti a scelta tra:								
inf0007	Analisi e Visualizzazione di Reti Complesse	B	C11, attività caratterizzante - discipline informatiche	inf/01	9	1 o 2	1 o 2	2 sem
mfn0943	Apprendimento Automatico e Analisi Intelligente dei Dati ANNI ALTERNI 15-16 NO	B	C11, attività caratterizzante - discipline informatiche	inf/01	9	2	2	2 sem
mfn0947	Basi di Dati Multimediali	B	C11, attività caratterizzante - discipline informatiche	inf/01	9	1o2	1 o 2	2 sem
mfn0942	Intelligenza Artificiale e Laboratorio	B	C11, attività caratterizzante - discipline informatiche	inf/01	9	1o2	1 o 2	2 sem
mfn0940	Modelli Avanzati e Architetture di Basi di Dati	B	C11, attività caratterizzante - discipline informatiche	inf/01	9	1	1	1 sem
mfn0949	Sistemi Cognitivi	B	C11, attività caratterizzante - discipline informatiche	inf/01	9	1o2	1 o 2	2 sem
mfn0959	Verifica dei Programmi Concorrenti	B	C11, attività caratterizzante - discipline informatiche	inf/01	9	1 o 2	1 o 2	2 sem
Un Insegnamento a scelta tra:								
mfn0974	Elaborazione Digitale Audio e Musica	B	C11, attività caratterizzante - discipline informatiche	inf/01	6	1	1	2 sem
mfn0960	Modelli Concorrenti e Algoritmi distribuiti	B	C11, attività caratterizzante - discipline informatiche	inf/01	6	1 o 2	1	1 sem
inf0008	Programmazione per Dispositivi Mobili	B	C11, attività caratterizzante - discipline informatiche	inf/01	6	1 o 2	1 o 2	2 sem
mfn0977	Reti Neurali	B	C11, attività caratterizzante - discipline informatiche	inf/01	6	1 o 2	2	1 sem
mfn0795	Sistemi di Calcolo Paralleli e Distribuiti	B	C11, attività caratterizzante - discipline informatiche	inf/01	6	1 o 2	1 o 2	2 sem
<i>Gli insegnamenti seguenti sono da tenere in considerazione in questo blocco solo per integrare argomenti non coperti completamente nella laurea di provenienza e comunque in base alle indicazioni della commissione.</i>								

mfn0985	Istituzioni di Linguaggi Formali	B	C11, attività caratterizzante - discipline informatiche	inf/01	6	1	1	1 sem
mfn0987	Istituzioni di Sistemi Intelligenti	B	C11, attività caratterizzante - discipline informatiche	inf/01	6	1	1	2 sem
mfn0989	Istituzioni di Sviluppo Software	B	C11, attività caratterizzante - discipline informatiche	inf/01	6	1	1	2 sem
mfn0990	Istituzioni di Tecnologie Web	B	C11, attività caratterizzante - discipline informatiche	inf/01	6	1	1	1 sem
Formazione Affine ed interdisciplinare:								
Due insegnamenti a scelta tra: (è fondamentale l'inserimento dell'insegnamento di Metodi Numerici, anche se questo compare in un gruppo di scelta, a meno che non siano state acquisite competenze equivalenti durante il percorso di studi triennale)								
mfn0971	Complementi di Analisi e Probabilità	C	A11, attività affini o integrative	3 mat/05, 3 mat/06	6	1	1	1 sem
mfn0970	Fisica per Applicazioni di Realtà Virtuale	C	A11, attività affini o integrative	fis/01	6	1	1	1 sem
mfn1001	Istituzioni di Economia e Gestione dell'Impresa	C	A11, attività affini o integrative	secs-p/08	6	1	1	2 sem
mfn0962	Metodi Numerici	C	A11, attività affini o integrative	mat/08	6	1	1	1 sem
Crediti Liberi RANGE 12-21 <i>(si possono scegliere gli insegnamenti che non siano stati precedentemente selezionati, partendo da questo o da altro curriculum, considerando che alcuni insegnamenti possono essere forniti anche in versione ridotta da 6 cfu)</i>								
Altre competenze:								
mfn0963	Lingua Inglese II	F	lettera d, altre conoscenze	L-Lin/12	3	1 o 2	1 o 2	1 e 2 sem
mfn0979	Prova Finale	E			24	2	2	

Curriculum “Reti e Sistemi informatici”

Il curriculum in Reti e Sistemi Informatici si propone di formare professionisti nel settore sistemistico dell'Informatica. I laureati potranno operare presso industrie, case produttrici di software, centri di ricerca ed enti pubblici.

In particolare, l'obiettivo primario di un curriculum è la formazione di figure professionali che siano in grado di ideare, sviluppare e gestire progetti informatici di sistemi complessi basati su componenti, applicazioni e servizi coordinati per mezzo di protocolli di comunicazione di livello strutturale ed applicativo, sempre più spesso distribuiti in rete.

Il curriculum intende offrire le competenze necessarie per affrontare i problemi posti dalla necessità di garantire un livello elevato di robustezza, affidabilità, continuità e qualità di servizio in sistemi spesso dotati di un crescente livello di autonomia ed ottimizzati per un ridotto consumo energetico. A seconda degli insegnamenti scelti i laureati di questo curriculum saranno dotati di ampie basi metodologiche nel campo dell'analisi e modellazione di sistemi, delle basi di dati e dei sistemi informativi, e disporranno di un'approfondita conoscenza ed esperienza delle metodologie, dei linguaggi e degli strumenti più avanzati nel campo della specifica, implementazione, testing, assicurazione di qualità e gestione di sistemi informatici distribuiti ad alta complessità. In particolare, essi saranno in grado di inserirsi rapidamente e professionalmente, con ruolo trainante anche di carattere dirigenziale, in tutti i processi produttivi relativi alla definizione, sviluppo o acquisizione, messa in campo e gestione di sistemi informatici integrati; di ideare e gestire servizi di rete di aziende e strutture pubbliche e private anche di grandi dimensione; di fornire supporto alle scelte della direzione in materia di automazione e di informatizzazione dei processi amministrativi e/o produttivi. La base culturale fornita a questi laureati permetterà loro di inserirsi in centri di ricerca e sviluppo pubblici e

privati, nonché in aziende, enti ed organismi che offrono e gestiscono servizi informatici e telematici, sia nazionali che internazionali.

Sfruttando gli insegnamenti a scelta, con l'ausilio dei docenti, lo studente sarà poi in grado di costruire curricula personalizzati che compongano queste competenze in un percorso orientato allo studio di grandi sistemi software distribuiti.

Articolazione del curriculum

Gli insegnamenti si dividono in:

- insegnamenti obbligatori (per 18 cfu) e insegnamenti opzionali in ristretto gruppo di scelta (per 33 cfu)
- insegnamenti opzionali in un vasto gruppo di scelta (18cfu) e insegnamenti liberi (12 cfu)
- insegnamenti affini e integrativi (12 cfu), inglese II (3 cfu) e prova finale (24 cfu) come per gli altri curricula.

Il piano di studio consigliato è riportato nella seguente tabella, in cui “Codice” e “Insegnamento” sono il codice di Ateneo e il titolo dell’insegnamento, “SSD” è il [settore scientifico disciplinare](#) della insegnamento, “CFU” sono i crediti formativi associati all’insegnamento, “Anno” è l’anno del piano carriera in cui è previsto l’insegnamento. Qualora l’anno previsto sia “1 o 2”, quindi a scelta dello studente, la colonna “Anno consigliato” indica l’anno utilizzato per la preparazione dell’orario del corso.

Curriculum “Reti e Sistemi informatici” – Piano coorte 2015

Codice	Insegnamento	tip	area	SSD	CFU	anno	su g g e r i t o	Sem
CURRICULUM Reti e Sistemi Informatici, percorso 103								
Blocco 1 Formazione caratterizzante:								
inf0039	Complementi di Reti e Sicurezza	B	C11, attività caratterizzante - discipline informatiche	inf/01	12	1	1	1 sem
inf0006	Gestione delle Reti	B	C11, attività caratterizzante - discipline informatiche	inf/01	6	2	2	2 sem
Blocco 2 Un Insegnamento a scelta tra:								
mfn0899	Valutazione delle prestazioni: Simulazione e Modelli	B	C11, attività caratterizzante - discipline informatiche	inf/01	9	1 o 2	2	2 sem
mfn0959	Verifica dei Programmi Concorrenti	B	C11, attività caratterizzante - discipline informatiche	inf/01	9	1 o 2	2	2 sem
Blocco 3 Due Insegnamenti a scelta tra:								
inf0007	Analisi e Visualizzazione di Reti Complesse	B	C11, attività caratterizzante - discipline informatiche	inf/01	9	2	2	2 sem
mfn0947	Basi di Dati Multimediali	B	C11, attività caratterizzante - discipline informatiche	inf/01	9	1 o 2	2	2 sem
mfn0941	Sviluppo Software per Componenti e Servizi Web	B	C11, attività caratterizzante - discipline informatiche	inf/01	9	1 o 2	2	1 sem
mfn0961	Teoria dell'Informazione	B	C11, attività caratterizzante - discipline informatiche	inf/01	9	1	1	2 sem
Blocco 4 Un Insegnamento a scelta tra:								
mfn0969	Architettura degli Elaboratori II	B	C11, attività caratterizzante - discipline informatiche	inf/01	6	1 o 2	2	1 sem
mfn0960	Modelli Concorrenti e Algoritmi distribuiti	B	C11, attività caratterizzante - discipline informatiche	inf/01	6	1 o 2	1	1 sem

mfn0795	Sistemi di Calcolo Paralleli e Distribuiti	B	C11, attività caratterizzante – discipline informatiche	inf/01	6	1 o 2	2	2 sem
Blocco 5	Tre Insegnamenti a scelta tra:							
	Eventuali insegnamenti non selezionati nel blocco precedente							
mfn0969	Architettura degli Elaboratori II	B	C11, attività caratterizzante - discipline informatiche	inf/01	6	1 o 2	2	1 sem
mfn0960	Modelli Concorrenti e Algoritmi distribuiti	B	C11, attività caratterizzante - discipline informatiche	inf/01	6	1 o 2	1	1 sem
mfn0795	Sistemi di Calcolo Paralleli e Distribuiti	B	C11, attività caratterizzante – discipline informatiche	inf/01	6	1 o 2	2	2 sem
	Altri insegnamenti (non sono selezionabili gli insegnamenti “...- parteA) se già inseriti nella versione da 9 CFU							
mfn0997	Algoritmi e Complessità – Parte A	B	C11, attività caratterizzante - discipline informatiche	inf/01	6	1 o 2	1	2 sem
inf0010	Apprendimento Automatico e Analisi Intelligente dei Dati – Parte A ANNI ALTERNI 15-16 NO	B	C11, attività caratterizzante - discipline informatiche	inf/01	6	2	2	2 sem
mfn0993	Basi di Dati Multimediali – Parte A	B	C11, attività caratterizzante - discipline informatiche	inf/01	6	1 o 2	2	2 sem
mfn0944	Laboratorio Avanzato di Basi Dati ANNI ALTERNI 15-16 SI	B	C11, attività caratterizzante - discipline informatiche	inf/01	6	1	1	2 sem
mfn0953	Modelli e Metodi per il Supporto alle Decisioni ANNI ALTERNI 15-16 SI	B	C11, attività caratterizzante - discipline informatiche	inf/01	6	1	1	2 sem
inf0008	Programmazione per Dispositivi Mobili	B	C11, attività caratterizzante – discipline informatiche	inf/01	6	2	2	2 sem
mfn0954	Reti Complesse	B	C11, attività caratterizzante - discipline informatiche	inf/01	6	2	2	2 sem
mfn0977	Reti Neurali	B	C11, attività caratterizzante - discipline informatiche	inf/01	6	1 o 2	2	1 sem
mfn0995	Sviluppo Software per Componenti e Servizi Web – Parte A	B	C11, attività caratterizzante - discipline informatiche	inf/01	6	1 o 2	2	1 sem
Inf0011	Teoria dell’Informazione – Parte A	B	C11, attività caratterizzante - discipline informatiche	inf/01	6	1	1	2 sem
mfn1361	Valutazione delle prestazioni: Simulazione e Modelli - Parte A	B	C11, attività caratterizzante - discipline informatiche	inf/01	6	1 o 2	2	2 sem
mfn1360	Verifica dei Programmi Concorrenti - Parte A	B	C11, attività caratterizzante - discipline informatiche	inf/01	6	1 o 2	2	2 sem
	Gli insegnamenti seguenti sono da tenere in considerazione in questo blocco solo per integrare argomenti non coperti completamente nella laurea di provenienza e comunque in base alle indicazioni della commissione.							
mfn0985	Istituzioni di Linguaggi formali	B	C11, attività caratterizzante – discipline informatiche	inf/01	6	1 o 2	1	1 sem
mfn0987	Istituzioni di Sistemi Intelligenti	B	C11, attività caratterizzante - discipline informatiche	inf/01	6	1 o 2	1	2 sem
mfn0989	Istituzioni di Sviluppo Software	B	C11, attività caratterizzante – discipline informatiche	inf/01	6	1 o 2	1	2 sem
mfn0990	Istituzioni di Tecnologie Web	B	C11, attività caratterizzante – discipline informatiche	inf/01	6	1 o 2	1	1 sem
mfn0945	Sicurezza I	B	C11, attività caratterizzante - discipline informatiche	inf/01	6	1 o 2	1	2 sem
	Formazione Affine ed interdisciplinare:							
	Due Insegnamenti a scelta tra:							
mfn0971	Complementi di Analisi e Probabilità	C	A11, attività affini o integrative	3 mat/05	6	1 o 2	1	1 sem

				, 3 mat/06				
mfn0970	Fisica per Applicazioni di Realtà Virtuale	C	A11, attività affini o integrative	fis/01	6	1 o 2	1	1 sem
mfn1001	Istituzioni di Economia e Gestione dell'Impresa	C	A11, attività affini o integrative	secs- p/08	6	1	1	2 sem
mfn0984	Istituzioni di Logica	C	A11, attività affini o integrative	mat/01	6	1	1	1 sem
mfn0962	Metodi Numerici	C	A11, attività affini o integrative	mat/08	6	1 o 2	1	1 sem
mfn1349	Ottimizzazione Combinatoria ANNI ALTERNI 15-16 NO	C	A11, attività affini o integrative	mat/09	6	2	2	2 sem
Crediti Liberi RANGE 12-21 <i>(si possono scegliere gli insegnamenti che non siano stati precedentemente selezionati, partendo da questo o da altro curriculum, considerando che alcuni insegnamenti possono essere forniti anche in versione ridotta da 6 cfu)</i>								
Altre competenze:								
mfn0963	Lingua Inglese II	F	lettera d, altre conoscenze	L- Lin/12	3	1 o 2	1 o 2	1 e 2 sem
mfn0979	Prova Finale	E			24	2	2	

Curriculum “Sistemi per il Trattamento dell’Informazione”

Il curriculum “*Sistemi per il Trattamento dell’Informazione*” ha come obiettivo la formazione di una figura professionale con elevate competenze informatiche in grado di svolgere il ruolo di progettista di sistemi informatici, curando in particolare gli aspetti di modellizzazione dei dati e della conoscenza, di analisi dei flussi informativi e dei processi decisionali, di sviluppo di capacità di risoluzione automatica di problemi, utilizzando le moderne tecnologie di sviluppo software e le tecnologie collegate ad Internet e al web. Il corso di laurea magistrale intende offrire le competenze necessarie per ideare, progettare, sviluppare e gestire sistemi informatici che soddisfino le crescenti esigenze di integrazione e interazione tra sistemi informativi complessi, che offrano la possibilità di analizzare, gestire ed accedere a grandi moli di dati (anche eterogenei), la capacità di permettere un accesso facile ed efficiente a varie categorie di utenti, di gestire grandi basi di conoscenza e offrano supporto per le decisioni e che siano in grado di operare con alto grado di autonomia.

I laureati magistrali acquisiscono una solida preparazione nel campo delle basi di dati, dei sistemi informativi, delle moderne metodologie di programmazione e sviluppo di sistemi software, dell’intelligenza artificiale e delle tecniche di sviluppo di servizi basati su Internet.

Inoltre, il corso di laurea magistrale dedica attenzione ai fondamenti dell’informatica, e alle problematiche di interazione uomo-macchina relative allo sviluppo di interfacce efficaci; inoltre, completa la formazione nel settore matematico.

Il diffondersi della Società della Conoscenza richiede specialisti informatici in grado di cooperare con esperti di altri settori. A tal fine la laurea magistrale, pur enfatizzando le competenze informatiche, cura la formazione interdisciplinare dando particolare rilievo al settore economico e gestionale, e coprendo anche altri settori come le scienze cognitive e la bioinformatica.

La rilevanza delle tematiche affrontate nel corso di laurea magistrale, come data warehouse, data mining, interoperabilità tra sistemi informativi, accesso a basi di dati (anche multimediali), gestione della conoscenza, servizi informativi personalizzati, servizi centrati su web, sistemi autonomi è ormai assodata in molti settori industriali e dei servizi e sta crescendo rapidamente nella pubblica amministrazione.

I laureati Magistrali potranno trovare impiego come progettisti in aziende produttrici di soluzioni e servizi informatici innovativi (software house e aziende di consulenza informatica, non necessariamente ristrette a quelle che operano in ambito nazionale), in centri di ricerca e sviluppo pubblici e privati, in aziende che utilizzano metodologie informatiche innovative (dal settore automotive, al settore aerospaziale e quello robotica), nonché in aziende, enti ed organismi che offrono e gestiscono servizi informatici e telematici avanzati (tra questi vanno annoverati gli istituti bancari e vasti settori della pubblica amministrazione).

La laurea magistrale apre la possibilità di avvio alla ricerca attraverso il Dottorato di Ricerca o altre attività formative avanzate sia in ambito locale che internazionale. Inoltre, le competenze acquisite durante il corso di laurea magistrale consentono di avviare attività professionali autonome, prospettiva che oggi trova sempre più spazio nel mondo del lavoro.

Articolazione del curriculum

Per raggiungere gli obiettivi didattici sopra descritti, il curriculum “Sistemi per il Trattamento dell’Informazione” offre un certo numero di insegnamenti (alcuni comuni anche ad altri curricula) che permettono di seguire un percorso coerente.

Il curriculum offre dei gradi di libertà al suo interno in quanto tra gli insegnamenti caratterizzanti sono considerati obbligatori solo quelli che formano la base culturale comune e condivisa del curriculum mentre c’è la possibilità di optare tra un certo numero di insegnamenti caratterizzanti per accentuare la preparazione verso una tematica specifica.

Il curriculum prevede anche il completamento della formazione con insegnamenti affini ed interdisciplinari (in area economica/gestionale e nel campo della matematica computazionale).

Il piano di studio consigliato è riportato nella seguente tabella, in cui “Codice” e “Insegnamento” sono il codice di Ateneo e il titolo dell’insegnamento, “SSD” è il [settore scientifico disciplinare](#) della insegnamento, “CFU” sono i crediti formativi associati all’insegnamento, “Anno” è l’anno del piano carriera in cui è previsto l’insegnamento. Qualora l’anno previsto sia “1 o 2”, quindi a scelta dello studente, la colonna “Anno consigliato” indica l’anno utilizzato per la preparazione dell’orario del corso.

Curriculum “Sistemi per il Trattamento dell’Informazione” –Piano coorte 2015

Codice	Insegnamento	tip	area	SSD	CFU	anno	suggerito	Sem
CURRICULUM Sistemi per il Trattamento dell'Informazione, percorso 104								
Formazione caratterizzante:								
mfn0942	Intelligenza Artificiale e Laboratorio	B	C11, attività caratterizzante - discipline informatiche	inf/01	9	1	1	2 sem
mfn0940	Modelli Avanzati e Architetture di Basi di Dati	B	C11, attività caratterizzante - discipline informatiche	inf/01	9	1	1	1 sem
mfn0941	Sviluppo Software per Componenti e Servizi Web	B	C11, attività caratterizzante - discipline informatiche	inf/01	9	1 o 2	2	1 sem
Due Insegnamenti a scelta tra:								
mfn0943	Apprendimento Automatico e Analisi Intelligente dei Dati ANNI ALTERNI 15-16 NO	B	C11, attività caratterizzante - discipline informatiche	inf/01	9	2	2	2 sem
mfn0947	Basi di Dati Multimediali	B	C11, attività caratterizzante - discipline informatiche	inf/01	9	1 o 2	1 o 2	2 sem
mfn0972	Elaborazione di Immagini e Visione Artificiale	B	C11, attività caratterizzante - discipline informatiche	inf/01	9	1 o 2	2	1 sem
mfn0949	Sistemi Cognitivi	B	C11, attività caratterizzante - discipline informatiche	inf/01	9	1 o 2	2	2 sem

Quattro Insegnamenti a scelta tra:								
mfn1348	Agenti Intelligenti	B	C11, attività caratterizzante - discipline informatiche	inf/01	6	1	1	2 sem
mfn0997	Algoritmi e Complessità – Parte A	D	altre attività, a scelta	inf/01	6	1 o 2	1 o 2	2 sem
mfn0969	Architettura degli Elaboratori II	B	C11, attività caratterizzante - discipline informatiche	inf/01	6	1 o 2	1 o 2	1 sem
mfn0951	Bioinformatica	B	C11, attività caratterizzante - discipline informatiche	inf/01	6	1 o 2	1 o 2	1 sem
mfn0944	Laboratorio Avanzato di Basi Dati ANNI ALTERNI 15-16 SI	B	C11, attività caratterizzante - discipline informatiche	inf/01	6	1	1	2 sem
mfn0960	Modelli Concorrenti e Algoritmi distribuiti	B	C11, attività caratterizzante - discipline informatiche	inf/01	6	1 o 2	1	1 sem
mfn0953	Modelli e Metodi per il Supporto alle Decisioni ANNI ALTERNI 15-16 SI	B	C11, attività caratterizzante - discipline informatiche	inf/01	6	1	1	2 sem
inf0008	Programmazione per Dispositivi Mobili	B	C11, attività caratterizzante - discipline informatiche	inf/01	6	1 o 2	2	2 sem
mfn0954	Reti Complesse	B	C11, attività caratterizzante - discipline informatiche	inf/01	6	1 o 2	2	2 sem
mfn0952	Sicurezza II	B	C11, attività caratterizzante - discipline informatiche	inf/01	6	1 o 2	1 o 2	1 sem
inf0040	Trattamento dei Dati e dei Processi Aziendali ANNI ALTERNI 15-16 NO	B	C11, attività caratterizzante - discipline informatiche	inf/01	6	2	2	2 sem
<i>Gli insegnamenti seguenti sono da tenere in considerazione in questo blocco solo per integrare argomenti non coperti completamente nella laurea di provenienza e comunque in base alle indicazioni della commissione.</i>								
mfn0939	Calcolabilità e Complessità B	B	C11, attività caratterizzante - discipline informatiche	inf/01	6	1 o 2	1 o 2	1 sem
mfn0986	Istituzioni di Interazione Uomo Macchina	B	C11, attività caratterizzante - discipline informatiche	inf/01	6	1	1	1 sem
mfn0988	Istituzioni di programmazione distribuita in rete	B	C11, attività caratterizzante - discipline informatiche	inf/01	6	1	1	1 sem
mfn0987	Istituzioni di Sistemi Intelligenti	B	C11, attività caratterizzante - discipline informatiche	inf/01	6	1	1	2 sem
mfn0989	Istituzioni di Sviluppo Software	B	C11, attività caratterizzante - discipline informatiche	inf/01	6	1	1	2 sem
mfn0990	Istituzioni di Tecnologie Web	B	C11, attività caratterizzante - discipline informatiche	inf/01	6	1	1	1 sem
mfn0945	Sicurezza I	B	C11, attività caratterizzante - discipline informatiche	inf/01	6	1 o 2	1 o 2	2 sem
Formazione Affine ed interdisciplinare:								
Due Insegnamenti a scelta tra:								
mfn0971	Complementi di Analisi e Probabilità	C	A11, attività affini o integrative	3 mat/05, 3 mat/06	6	1 o 2	1 o 2	1 sem
mfn0946	Economia e Gestione delle Imprese Net Based	C	A12, attività affini o integrative	secs-p/08	6	1 o 2	1	2 sem
mfn0970	Fisica per Applicazioni di Realtà Virtuale	C	A11, attività affini o integrative	fis/01	6	1	1	1 sem
mfn0962	Metodi Numerici	C	A12, attività affini o integrative	mat/08	6	1	1	1 sem
mfn1349	Ottimizzazione Combinatoria ANNI ALTERNI 15-16 NO	C	A11, attività affini o integrative	mat/09	6	2	2	2 sem
Crediti Liberi RANGE 12-21								

	<i>(si possono scegliere gli insegnamenti che non siano stati precedentemente selezionati, partendo da questo o da altro curriculum, considerando che alcuni insegnamenti possono essere forniti anche in versione ridotta da 6 cfu)</i>							
	Altre competenze:							
mfn0963	Lingua Inglese II	F	lettera d, altre conoscenze	L- Lin/12	3	1 o 2	1 o 2	1 e 2 sem
mfn0979	Prova Finale	E			24	2	2	

Dopo la magistrale: il dottorato di ricerca

Scopo fondamentale del Corso di Dottorato è l'acquisizione del metodo di indagine scientifica, che dovrà essere dimostrata con la stesura di una tesi di ricerca contenente risultati originali conseguiti in un settore dell'Informatica.

Al dottorato si accede tramite esami di ammissione ed il dottorato è a numero chiuso. Per un certo numero di posizioni sono previste borse di dottorato finanziate dagli Atenei e dal Ministero. Borse addizionali sono finanziate su fondi di Enti, Fondazioni ed industrie. Il dottorato di Informatica è parte della Scuola di Dottorato in "[Science and Advanced Technology](#)" dell'Università di Torino.

Il Dottorato di Ricerca in Informatica presso il Dipartimento di Informatica

Il Dottorato di Ricerca in Informatica è ad ampio spettro e copre le principali aree dell'informatica, con particolare riferimento alle aree di ricerca attive presso il dipartimento di Informatica (<http://di.unito.it/ricerca>).

Al Dottorato si accede attraverso esame di ammissione. La domanda viene presentata di norma nell'estate, in risposta a specifico bando. Il bando prevede la possibilità di iscrizione 'sub condizione' all'esame di ammissione al dottorato per i laureandi che conseguiranno la laurea nella sessione autunnale.

Organizzazione del Dottorato

Il Dottorato di Informatica è articolato in tre anni e coordinato dal Collegio dei Docenti, che include docenti delle varie aree scientifiche del Dipartimento di Informatica. Il Collegio dei Docenti, tenendo anche conto delle eventuali indicazioni dei dottorandi, assegna ad ognuno di essi un tutore scelto fra i membri del Collegio. Gli studenti di dottorato, con il concorso del tutore, formulano un piano di studio che deve essere approvato dal Collegio dei Docenti.

Nel corso del primo e del secondo anno i dottorandi approfondiscono lo studio di uno o più settori dell'informatica al fine di acquisire le conoscenze necessarie per la tesi di ricerca.

Pertanto ogni studente di dottorato deve frequentare sei corsi (ciascuno tipicamente di almeno 20 ore di lezione) e superare i relativi esami (sono possibili debiti formativi per quegli studenti di dottorato che non hanno avuto modo di approfondire la loro preparazione in alcuni settori dell'informatica durante il conseguimento della laurea).

La preparazione della tesi di dottorato è l'attività prevalente nel secondo e nel terzo anno di dottorato. Nel formulare la valutazione finale della attività del candidato al termine del terzo anno, il Collegio dei Docenti tiene conto della valutazione della tesi fatta da due revisori (di cui almeno uno straniero) scelti dal Collegio stesso tra ricercatori esperti nella particolare sottoarea in cui è stata svolta la tesi di dottorato.

Informazioni

Il Coordinatore del dottorato di ricerca in Informatica è la prof.ssa Mariangiola Dezani (dezani@di.unito.it). Informazioni dettagliate ed aggiornate sulla attività del dottorato di Informatica e della Scuola di Dottorato in Scienza ed Alta Tecnologia sono reperibili alla pagina <http://dott-sat.campusnet.unito.it/>

Supporto agli studi

Erasmus

L'Università di Torino ha stabilito accordi bilaterali con un certo numero di Università europee per lo scambio di studenti di Informatica nel quadro del progetto Socrates/Erasmus dell'Unione Europea. Gli studenti del Corso di Laurea Magistrale in Informatica possono quindi concorrere a borse annuali per un soggiorno di studio all'estero, con convalida degli esami colà sostenuti.

Gli accordi attualmente esistenti riguardano:

Universidad Autonoma de Barcelona, Spagna
Universidad de Las Palmas de Gran Canaria, Isole Canarie Spagna
Universidad Autonoma de Madrid, Spagna
Universidad Politecnica de Madrid, Spagna
Universidad de La Laguna, Isole Canarie-Tenerife Spagna
Universidad de Zaragoza, Spagna
Universite de Savoie, Francia
Université de Lorraine, Nancy, Francia
Université Paris Diderot – Paris 7, Francia
Reykjavik University, Islanda
Technical University of Lodz, Polonia
Universitatea “Politehnica” din Bucuresti, Romania
Universitatea Tehnica Cluj-Napoca – Technical University of Cluj-Napoca, Romania
Université du Luxembourg, Lussemburgo
Universitat Klagenfurt, Austria
Universite Paris Nord – Paris XIII, Francia
Université Pierre et Marie Curie, Paris 6, Francia (per Erasmus Traineeship)
University of Helsinki, Finlandia
Sabaci University, Istanbul, Turchia
Université Montpellier 2, Francia
Wroclaw University of Technology, Polonia
Universidad Politecnica de Valencia, Spagna

Altre informazioni sono disponibili sui siti <http://di.unito.it/erasmus> e <http://di.unito.it/erasmusccs>

Biblioteca

Come molte biblioteche universitarie, anche la biblioteca del Dipartimento di Informatica (<http://di.unito.it/biblio>) contiene collezioni di interesse sia per gli studenti che per i ricercatori.

Per quanto riguarda l'aspetto didattico la biblioteca dispone di buona parte dei testi di studio relativi ai diversi insegnamenti. Molti insegnamenti, accanto ad un testo fondamentale o a una raccolta di dispense, consigliano altri libri complementari: la biblioteca offre la possibilità di consultare tali testi e di ottenerne il prestito per lo studio individuale.

Il patrimonio documentario è costituito da numerosi testi monografici e riviste specialistiche, in parte accessibili a scaffale aperto, in parte accessibili su richiesta del personale bibliotecario. La

biblioteca dispone di una sala studio-consultazione dove è anche possibile effettuare ricerche bibliografiche online dai cataloghi accessibili da due postazioni informatiche.

Per comunicare con la biblioteca: biblioteca@di.unito.it

Per maggiori informazioni e dettagli su orari e servizi, consultare il sito <http://di.unito.it/biblioteca>.

Supporto on-line agli insegnamenti

Oltre ai libri di testo e ad eventuali dispense, i docenti degli insegnamenti utilizzano una [piattaforma di e-learning](#) per la distribuzione di ulteriore materiale didattico, per la consegna di eventuali esercizi, la distribuzione di esercizi di autovalutazione, e blog di discussione fra studenti dello stesso insegnamento.

Aule e laboratori

Le lezioni si svolgono nelle aule e nei laboratori situati in Via Pessinetto 12 a Torino, presso la sede del Dipartimento di Informatica dell'Università degli Studi di Torino.

I laboratori didattici (<http://di.unito.it/laboratori>) offrono oltre 140 postazioni di lavoro (personal computer e workstation) organizzate in cinque aree (laboratorio Turing, laboratorio Von Neumann, laboratorio Dijkstra, laboratorio Babbage e laboratorio Postel). Nei primi quattro si svolgono le lezioni degli insegnamenti di laboratorio, mentre il laboratorio Postel è principalmente dedicato allo sviluppo delle tesi di laurea. Nei periodi in cui i laboratori non sono occupati da lezioni, le postazioni di lavoro sono a disposizione degli studenti per esercitazioni individuali e per la preparazione della tesi. Ogni laboratorio è organizzato come rete locale, e queste reti sono collegate fra loro in configurazione intranet.

Ad ogni studente viene assegnato un codice personale di accesso alle macchine, indispensabile per l'uso delle postazioni di lavoro. Questo codice di accesso è differente dalle vostre credenziali di Ateneo, e serve solo per l'accesso ai laboratori del Dipartimento di Informatica. I laboratori sono a disposizione degli studenti per i soli scopi didattici, e vengono gestiti nella più stretta osservanza delle norme contro la pirateria informatica.

Gli studenti sono invitati a richiedere il più presto possibile, dopo l'iscrizione, il codice di accesso alle macchine (login), rivolgendosi ai tecnici del Dipartimento di Informatica negli orari fissati (<http://di.unito.it/login>).

I laboratori sono aperti dal Lunedì al Venerdì dalle ore 8.30 alle ore 19:00.

Zone studio

Gli studenti possono utilizzare un'ampia zona studio messa a disposizione dall'Ente Diritto allo Studio Universitario della Regione (EDISU), in locali contigui alle aule.

Supporto per gli studenti disabili

L'Università degli Studi di Torino, nella prospettiva di rendere effettivo il diritto allo studio per tutti gli studenti disabili, intende garantire l'accesso fisico alle strutture di studio e di ricerca. Esiste un progetto di progressiva eliminazione delle barriere architettoniche che, ogni anno, disponendo di apposita quota di finanziamento, affronta le situazioni che sono state individuate, attraverso un censimento di tutti gli edifici, come maggiormente problematiche e gravi.

Gli studenti disabili possono usufruire di agevolazioni relative al pagamento delle tasse in relazione alla percentuale di disabilità.

Gli studenti disabili possono, inoltre, richiedere diverse tipologie di servizi:

- accompagnamento presso le strutture universitarie e gli enti di ricerca ed assistenza durante i pasti;
- tutorato didattico: aiuto per la compilazione di appunti, il reperimento testi, fotocopie, disbrigo pratiche burocratiche;
- supporto per la preparazione degli esami (rivolto esclusivamente a studenti con particolari disabilità);
- supporto di interpreti della Lingua Italiana dei Segni e di Mediatori alla Comunicazione per gli studenti non udenti;
- supporto per la richiesta di prove d'esame individualizzate;
- possibilità di utilizzo dei locali del Settore per attività di studio connesse all'Ateneo e disponibilità di postazioni informatiche accessibili;
- sostegno personalizzato attraverso progetti individuali specifici.

Gli studenti disabili possono inoltre accedere ai servizi dell'Ente per il Diritto allo Studio Universitario della Regione Piemonte e del Settore Mobilità Internazionale secondo le modalità individuate dai bandi di concorso.

L'elenco dei Delegati per gli studenti disabili delle Scuole e dei Dipartimenti è disponibile sul sito di Ateneo www.unito.it » Servizi per gli studenti » Studenti disabili » Delegati

Per informazioni

Direzione Risorse Umane

Settore Integrazione Studenti Disabili

Via Po, 31 – Via Po, 29 (ingresso studenti) – 10124 Torino

Tel. 011.670.4282/4283/4284 – Fax 011.670.4285 – Email: ufficio.disabili@unito.it

Orari:

- apertura dei locali dedicati agli studenti per attività di studio e di ricerca, incontri con i propri operatori e utilizzo di postazioni informatiche accessibili, preferibilmente su prenotazione, da lunedì a giovedì 9-18.45 e venerdì 9-16.30;
- ricevimento studenti per colloqui specifici e/o prima accoglienza esclusivamente su appuntamento.

La Scuola di Scienze della Natura, a cui il nostro corso di Laurea Magistrale afferisce, ha nominato come garante per gli studenti disabili della propria Scuola la prof.ssa Anna Capietto, a cui è possibile rivolgersi per informazioni specifiche sui test d'ingresso, sulla modalità

d'esame, sui percorsi didattici specifici e particolari. Referente per il nostro Corso di Laurea Magistrale è la prof.ssa Cristina Baroglio (supplente Prof. Luca Padovani).

Prof.ssa **Capietto Anna** tel 0116702914 fax 0116702878 anna.capietto@unito.it Dip. Matematica "Giuseppe Peano" Via Carlo Alberto, 10 – 10123 Torino

Prof.ssa **Baroglio Cristina**, tel. 0116706711, cristina.baroglio@unito.it, Dipartimento di Informatica, Via Pessinetto 12, Torino
(supplente Prof. **Padovani Luca**, tel. 0116706711, luca.padovani@unito.it, Dipartimento di Informatica, Via Pessinetto 12, Torino)

Tutte le informazioni sono reperibili al seguente link:

- studenti con disabilità: <http://www.unito.it/servizi/lo-studio/studenti-disabili>

Supporto per gli studenti con DSA (Disturbi Specifici di Apprendimento)

L'Università di Torino ha uno **sportello dedicato** agli studenti con DSA. L'Università di Torino, in sintonia con le recenti disposizioni di legge, in specifico la legge 8 ottobre 2010, n. 170, si impegna ad adottare modalità di apprendimento e di valutazione il più possibile flessibili, in grado di **garantire il pieno apprendimento** in termini di conoscenze e competenze anche per gli studenti DSA. Lo sportello predisposto offre i seguenti servizi:

- interventi di **mediazione** con i docenti in vista degli esami orali o scritti;
- **tutorato specifico** (redazione appunti, registrazione lezioni) per le attività didattiche;
- **informazioni** sulle procedure di immatricolazione e sui test d'ingresso;
- incontri individuali di **consulenza didattica**.

L'elenco dei Referenti per gli studenti con DSA è disponibile sul sito di Ateneo www.unito.it » Servizi per gli studenti » Studenti dislessici » Referenti

Per informazioni

Direzione Risorse Umane

Sportello Dislessia

Via Po, 31 – 10124 Torino

Tel. 011.670.4282/4283/4284 – Fax 011.670.4285 – Email: sportello.dislessia@unito.it

www.unito.it » Servizi per gli studenti » Studenti dislessici

Referente per il nostro Corso di Laurea è la prof.ssa Cristina Baroglio (supplente Prof. Luca Padovani).

Prof.ssa **Baroglio Cristina**, tel. 0116706711, cristina.baroglio@unito.it, Dipartimento di Informatica, Via Pessinetto 12, Torino
(supplente Prof. **Padovani Luca**, tel. 0116706711, luca.padovani@unito.it, Dipartimento di Informatica, Via Pessinetto 12, Torino)

Tutte le informazioni sono reperibili al seguente link:

- studenti con DSA: <http://www.unito.it/servizi/lo-studio/studenti-dislessici>

Indirizzi utili

Ufficio del Corso di Laurea e Laurea Magistrale in Informatica

*C.so Svizzera 185, Torino (ingresso da Via Pessinetto 12 – I piano)
da Lunedì a Venerdì solo su appuntamento**

e-mail: informatica@educ.di.unito.it

Telefono: 011 – 670 67 41; 011 – 670 68 25; fax 011 - 751603

**per appuntamento scrivere a: informatica@educ.di.unito.it con almeno 2 giorni lavorativi d'anticipo*

Segreteria Studenti (<http://di.unito.it/segreteriaStudenti>)

Via Santa Croce, 6 – Torino

Lun e Ven: 9-11; Mar, Mer, Gio: 9-11 e 13.30-15

e-mail: segreteriamfn@unito.it

Telefono: 011 – 670 46 29/30/31/32/33/34; fax 011 – 670 46 93

Job Placement (<http://di.unito.it/jobplacement>)

Via Michelangelo 32 – Torino

Da Lun a Ven: 09.30-12 e 13.30-16

e-mail: jp.scienzedellanatura@unito.it

Telefono: 011 – 670 6215-6216-6218-6219; fax: 011 – 670 6217

Presidente del Consiglio dei Corsi di Laurea e Laurea Magistrale in Informatica: prof. Susanna Donatelli

e-mail: presccs@educ.di.unito.it

Direttore del Dipartimento di Informatica: prof. Luca Console

e-mail: direttore@di.unito.it

Ufficio Accertamento Economico, Regolamento Tasse, inserimento Fasce di reddito
(<http://di.unito.it/tasse>)

Vicolo Benevello 3/a, Torino

Apertura sportelli: Lun e Ven: 9-11; Mar, Mer, Gio: 9-11 e 13.30-15

Tel. 011 – 670 9902 (solo da lun a ven ore 9.00-12.00); fax 011 – 23 610 73

Ente Diritto allo Studio Universitario (<http://www.edisu.piemonte.it>)

(borse di studio, posto letto, buoni mensa)

Via Giulia di Barolo, 3/bis – 10124 Torino

Tel. 011 – 652 27 01 E-mail: edisu@edisu-piemonte.it

Lun e Ven: 9-11; Mar, Mer, Gio: 9-11 e 13.30-15

Ufficio Studenti Stranieri (<http://di.unito.it/stranieri>)

Vicolo Benevello, 3/a (I piano) – 10124 Torino

Lun e Ven: 9-11; Mar, Mer, Gio: 9-11 e 13.30-15

e-mail: segrstu.stranieri@unito.it

Telefono: 011 – 670 4498 oppure 011 – 670 4499

Ufficio Erasmus (<http://di.unito.it/erasmus>)

Vicolo Benevello 3/A – 10124 Torino

Lun e Ven: 9-11; Mar, Mer, Gio: 9-11 e 13.30-15

Per gli Studenti: relint@unito.it

Telefono: 011 – 670 4425

Rappresentanti degli Studenti: mail rappstud@di.unito.it; <http://studenti.i-learn.unito.it/>

Seconda Parte: Programmi e docenti degli insegnamenti per l’A.A. 15/16

In questa seconda parte della guida vengono riportati tutti gli insegnamenti attivati nell’anno accademico 15/16 presso il Corso di Laurea Magistrale di Informatica, di interesse per gli studenti immatricolati nell’anno e per gli studenti della coorte precedente che devono seguire gli insegnamenti del secondo anno. Sono inoltre riportati gli insegnamenti che sono tenuti in mutuaione dalla triennale, per il recupero di competenze. Infine, per facilitare la compilazione del piano carriera degli studenti della coorte 2015, questa seconda parte della guida riporta anche un elenco degli insegnamenti che non verranno tenuti nel 15/16 ma che saranno attivi nel 16/17 (vuoi perché sono insegnamenti previsti ad anni alterni, vuoi perché si tratta di insegnamenti di nuova istituzione l’attivazione dei quali è già stata deliberata ma che, non essendo del primo anno, non appartengono propriamente all’offerta formativa 15/16).

Insegnamenti attivi nel 15/16.

La tabella che segue elenca tutti gli insegnamenti che potrete seguire nell’A.A. 15/16. Se un insegnamento elencato nell’offerta formativa della vostra coorte non è presente in questa tabella, allora vuol dire che l’insegnamento sarà attivo nel prossimo A.A., vuoi perché si tratta di un insegnamento di nuova attivazione che è previsto al secondo anno della vostra coorte (e quindi nel 16/17 nessun studente potrebbe averlo nel libretto on-line), vuoi perché si tratta di un insegnamento che viene offerto solo ad anni alterni, quindi attenzione a seguirlo nell’anno in cui è effettivamente offerto. Nella tabella “Codice” e “Insegnamento” sono il codice di Ateneo e il titolo dell’insegnamento, “SSD” è il [settore scientifico disciplinare](#) dell’insegnamento, “CFU” sono i crediti formativi associati all’insegnamento, “docenti” sono i docenti responsabili dell’insegnamento e “semestre” il periodo didattico in cui si tiene l’ insegnamento. I programmi degli insegnamenti sono riportati in appendice. Per ulteriori informazioni potete visitare le pagine web del corso di laurea magistrale (da <http://magistrale.educ.di.unito.it/> seguire il link ‘insegnamenti’, nella sezione “Per chi studia con noi”).

Codice principale	INSEGNAMENTO	SSD	TAF	Ambito	CFU	DOCENTI	Semestre	NOTE
mfn1348	Agenti Intelligenti	inf/01	B	caratterizzante	6	Martelli	2	
mfn0997	Algoritmi e Complessità A	inf/01	B	caratterizzante	6	Zacchi	2	
Inf0007	Analisi e Visualizzazione di Reti Complesse	inf/01	B	caratterizzante	9	Ruffo, visiting professor Bollen	2	
mfn0943	Apprendimento Automatico e Analisi Intelligente dei Dati	inf/01	B	caratterizzante	9	Meo, Esposito	(2)	Ad anni alterni, non si terrà nel 15-16
mfn0969	Architettura degli Elaboratori II	inf/01	B	caratterizzante	6	Gunetti, Aldinucci	1	
mfn0947	Basi di Dati Multimediali	inf/01	B	caratterizzante	9	Sapino	2	
mfn0951	Bioinformatica	inf/01	B	caratterizzante	6	Botta Calogero	1	
mfn0939	Calcolabilità e Complessità B	inf/01	B	caratterizzante	6	Ronchi della Rocca	1	per il 15-16 mutuato su Metodi Formali dell’Informatica, triennale 270.
mfn0971	Complementi di Analisi e Probabilità	3 mat/05,	C	affine e integrative	6	Bibbona/Boggia tto	1	

		3 mat/06						
inf0039	Complementi di Reti e Sicurezza	inf/01	B	caratterizzante	12	Bergadano (sicurezza), Garetto (Reti)	1	
mfn0946	Economia e Gestione delle Imprese Net Based	secs-p/08	C	affine e integrative	6	Pironti/Pisano	2	
mfn0972	Elaborazione di Immagini e Visione Artificiale	inf/01	B	caratterizzante	9	Balossino, Cavagnino, Grangetto	1	
mfn0974	Elaborazione Digitale Audio e Musica	inf/01	B	caratterizzante	6	Lombardo, Valle	2	mutuato su insegnamenti presso DAMS (Informazioni alla pagina: http://www.di.unito.it/~vincenzo/ElabSuoMus (sede: DAMS-Palazzo Nuovo, Via Verdi Torino)
mfn0970	Fisica per Applicazioni di Realtà Virtuale	fis/01	C	affine e integrative	6	Maggiora	1	
inf0006	Gestione delle Reti	inf/01	B	caratterizzante	6	Manini	2	
mfn0942	Intelligenza Artificiale e Laboratorio	inf/01	B	caratterizzante	9	Torasso, Pozzato	2	
mfn0944	Laboratorio Avanzato di Basi di Dati	inf/01	B	caratterizzante	6	Meo, Masera	2	
mfn0963	Lingua Inglese II	L-Lin/12	E	conoscenza lingua straniera	3	Griffin (esercitatore), Radicioni, Cordero, Zacchi responsabili	1 e 2	
mfn0962	Metodi Numerici	mat/08	C	affine e integrative	6	Semplice	1	
mfn0973	Modellazione Grafica	inf/01	B	caratterizzante	9	Balossino, Cavagnino, Faletti	2	
mfn0940	Modelli Avanzati e Architetture di Basi di Dati	inf/01	B	caratterizzante	9	Sacco	1	
mfn0960	Modelli Concorrenti e Algoritmi distribuiti	inf/01	B	caratterizzante	6	Margaria	1	
mfn0953	Modelli e Metodi per il Supporto alle Decisioni	inf/01	B	caratterizzante	6	Aringhieri, Grosso	2	Ad anni alterni, si terrà nel 15-16
mfn1349	Ottimizzazione Combinatoria	mat/09	C	affine e integrative	6	Aringhieri, Grosso	(2)	Ad anni alterni, non si terrà nel 15-16
inf0005	Programmazione per Dispositivi Mobili	inf/01	B	caratterizzante	6	Damiani	2	
mfn0954	Reti Complesse	inf/01	B o D	Caratterizzante o a scelta	6	Ruffo	2	mutuato da modulo di Analisi e Visualizzazione di Reti Complesse
inf0009	Reti II	inf/01	D	a scelta	6	Garetto	1	mutuato da modulo di Complementi di Reti e Sicurezza
mfn0977	Reti Neurali	inf/01	B	caratterizzante	6	Gliozzi	1	
mfn0952	Sicurezza II	inf/01	B o D	Caratterizzante o a scelta	6	Bergadano	1	mutuato da modulo di Complementi di Reti e Sicurezza
mfn0949	Sistemi Cognitivi	inf/01	B	caratterizzante	9	Boella, Mazzei, Radicioni	2	
mfn0795	Sistemi di Calcolo Paralleli e Distribuiti	inf/01	B	caratterizzante	6	Aldinucci	2	mutuato su insegnamento presso Laurea Magistrale in

							Fisica. Informazioni alla pagina http://fisica.campusnet.unito.it/do/corsi.pl/Show?_id=48d4;sort=DEFAULT;search=;hits=126 oppure http://fisica.campusnet.unito.it/do/lezioni.pl
mfn0978	Sistemi di Realtà Virtuale	inf/01	B	caratterizzante	9	Balossino, Grangetto, Lucenteforte	1
mfn0941	Sviluppo Software per Componenti e Servizi Web	inf/01	B	caratterizzante	9	Petrone	1
mfn0961	Teoria dell'Informazione	inf/01	B	caratterizzante	9	Grangetto, Sereno	2
inf0040	Trattamento dei Dati e dei Processi Aziendali	inf/01	B	caratterizzante	6		(2) Ad anni alterni, non si terrà nel 15-16
mfn0899	Valutazione delle prestazioni: Simulazione e Modelli	inf/01	B	caratterizzante	9	Balbo	2
mfn0959	Verifica dei Programmi Concorrenti	inf/01	B	caratterizzante	9	Donatelli	1
mfn0979	Prova finale	altre attività	E		24		

Insegnamenti mutuati attivi nell'A.A. 15/16

Gli insegnamenti mutuati sono insegnamenti con un codice proprio, ma per i quali non è previsto un insegnamento proprio, bensì una mutuazione da tutto o parte di un altro insegnamento. Vengono inseriti con due scopi principali: recupero di competenze utili della triennale oppure versione “corta”, cioè da 6 cfu, di un insegnamento da 9 cfu della magistrale. Nel primo caso gli insegnamenti tipicamente si chiamano “istituzioni di XX” e mutuano sull’insegnamento di nome “XX” della Laurea (triennale), possono essere inseriti in piano di studi solo in accordo con la commissione ammissioni. Nel secondo caso tipicamente si chiamano “XX – parte A” e mutuano 6 dei 9 cfu dell’ insegnamento “XX”) e possono essere inseriti in modo autonomo dallo studente in accordo con le regole dei piani carriera, normalmente nel gruppo degli insegnamenti liberi o in precisi gruppi di scelta, a patto che nel piano di studi dello studente non sia già presente la versione da 9 cfu.

Codice principale	INSEGNAMENTO	SSD	CFU	INSEGNAMENTO DA CUI MUTUA
Insegnamenti del primo tipo (taf B – caratterizzante oppure D – a scelta)				
mfn0939	Calcolabilità e Complessità B	inf/01	6	per il 15-16 vedi Metodi Formali dell'Informatica, triennale 270 su cui mutua (modulo da 6 cfu)
mfn1476	Istituzioni di Basi di Dati	inf/01	6	vedi Basi di Dati, triennale DM270, su cui mutua (modulo da 6 cfu)
mfn1477	Istituzioni di Basi di Dati	inf/01	9	vedi Basi di Dati, triennale DM270, su cui mutua
mfn1473	Istituzione di Calcolo Matriciale e Ricerca Operativa	Mat/09	6	vedi Calcolo Matriciale e Ricerca Operativa, triennale DM270, su cui mutua

mfn0985	Istituzione di Linguaggi formali	inf/01	6	vedi Linguaggi Formali e Traduttori, triennale DM270, su cui mutua (modulo da 6 cfu)
mfn1001	Istituzioni di Economia e Gestione dell'Impresa	secs-p/08	6	vedi Economia e Gestione dell'Impresa e Diritto, triennale DM270, su cui mutua (modulo da 6 cfu)
mfn0986	Istituzioni di Interazione Uomo Macchina	inf/01	6	vedi Interazione Uomo Macchina e Tecnologie Web, triennale DM270, su cui mutua (modulo da 6 cfu)
mfn0984	Istituzioni di logica	mat/01	6	vedi Matematica Discreta e Logica, triennale DM270, su cui mutua (modulo da 6 cfu)
mfn0988	Istituzioni di programmazione distribuita in rete	inf/01	6	vedi Programmazione III, triennale DM270, su cui mutua
mfn0987	Istituzioni di Sistemi Intelligenti	inf/01	6	vedi Sistemi Intelligenti, triennale DM270, su cui mutua
mfn0974	Istituzioni di Sistemi Operativi	inf/01	6	vedi Sistemi Operativi, triennale DM270, su cui mutua (modulo da 6 cfu)
mfn0975	Istituzioni di Sistemi Operativi	inf/01	12	vedi Sistemi Operativi, triennale DM270, su cui mutua
mfn0989	Istituzioni di Sviluppo Software	inf/01	6	vedi Sviluppo di applicazioni Software, triennale DM270, su cui mutua (modulo da 6 cfu)
mfn0990	Istituzioni di Tecnologie Web	inf/01	6	vedi Interazione Uomo Macchina e Tecnologie Web, triennale DM270, su cui mutua (modulo da 6 cfu)
mfn0945	Sicurezza I	inf/01	6	Vedi Sicurezza, triennale 270, su cui mutua
Insegnamenti del secondo tipo (taf B – caratterizzante, oppure D – a scelta)				
inf0010	Apprendimento Automatico e Analisi Intelligente dei Dati – Parte A	inf/01	6	Vedi Apprendimento Automatico e Analisi Intelligente dei Dati, cod. mfn0943, su cui mutua
mfn0993	Basi di Dati Multimediali – Parte A	inf/01	6	Vedi Basi di Dati Multimediali, cod. mfn0947, su cui mutua
mfn0991	Modelli Avanzati e Architetture di Basi di Dati – Parte A	inf/01	6	Vedi Modelli Avanzati e Architetture di Basi di Dati, cod. mfn0940, su cui mutua
mfn0954	Reti Complesse	inf/01	6	Vedi Analisi e Visualizzazione di Reti Complesse su cui mutua (modulo da 6 cfu)
inf0009	Reti II	inf/01	6	Vedi Complementi di Reti e Sicurezza su cui mutua (modulo da 6 cfu)
mfn0952	Sicurezza II	inf/01	6	Vedi Complementi di Reti e Sicurezza su cui mutua (modulo da 6 cfu)
mfn0995	Sviluppo Software per Componenti e Servizi Web – Parte A	inf/01	6	Vedi Sviluppo Software per Componenti e Servizi Web, cod. mfn0941, su cui mutua
inf0011	Teoria dell'Informazione – Parte A	inf/01	6	Vedi Teoria dell'Informazione, cod. mfn0961, su cui mutua
mfn1361	Valutazione delle prestazioni: Simulazione e Modelli - Parte A	inf/01	6	Vedi Valutazione delle prestazioni: Simulazione e Modelli, cod. mfn0899, su cui mutua
mfn1360	Verifica dei Programmi Concorrenti - Parte A	inf/01	6	Vedi Verifica dei Programmi Concorrenti , cod. mfn0959 – magistrale RETI, su cui mutua

Altri insegnamenti mututati (taf B – caratterizzante)			
mfn0974	Elaborazione Digitale Audio e Musica	inf/01	6
mutuato su insegnamenti presso DAMS (Informazioni alla pagina: http://www.di.unito.it/~vincenzo/ElabSuoMus (sede: DAMS-Palazzo Nuovo, Via Verdi Torino)			
mfn0795	Sistemi di Calcolo Paralleli e Distribuiti	inf/01	6
mutuato su insegnamento presso Laurea Magistrale in Fisica. Informazioni alla pagina http://fisica.campusnet.unito.it/do/corsi.pl/Show?_id=48d4;sort=DEFAULT;search=;hits=126 oppure http://fisica.campusnet.unito.it/do/lezioni.pl			

Insegnamenti ad anni alterni non tenuti nel 15/16:

Nell'AA 16/17, in coincidenza del secondo anno della coorte 2015, saranno attivati i seguenti insegnamenti:

- Apprendimento Automatico e Analisi Intelligente dei Dati, codice mfn0943, nel SSD inf/01 (anche la parte A)
- Ottimizzazione Combinatoria, codice mfn1349, nel SSD mat/09
- Trattamento dei Dati e dei Processi Aziendali, codice inf0040, nel SSD inf/01

Insegnamenti esterni all'Ateneo.

Il Corso di Laurea Magistrale ha un accordo di collaborazione con il Collegio Einaudi per il riconoscimento di corsi professionalizzanti tenuti presso la loro struttura. Lo studente che voglia seguire tali corsi, e che anche solo voglia tenersi aperta tale possibilità, per poi chiederne il riconoscimento, deve preventivamente inserire nel Piano Carriera i codici MFN1052 Attività Professionalizzanti – 3cfu, taf D e/o MFN1502 Attività Professionalizzanti II – 3cfu, taf D (*oltre* agli insegnamenti scelti come liberi – almeno 12 cfu). Solo in questo modo sarà possibile effettuare l'eventuale convalida: al momento della domanda di laurea, ciò che è stato utilizzato potrà essere depennato senza alcun costo.

mfn1052	Attività Professionalizzanti	altre attività, a scelta lettera A	3
mfn1502	Attività Professionalizzanti	altre attività, a scelta lettera A	3

Informazioni aggiornate al 29 Febbraio 2016

Programmi e altre informazioni per gli insegnamenti attivi nel 15/16 (syllabus degli insegnamenti).

Le informazioni saranno pubblicate nell'aggiornamento di settembre della guida, come previsto dal regolamento di Ateneo.

Nel frattempo gli studenti possono fare riferimento alle pagine dei corsi dell'A.A. 15/16 sul sito della laurea magistrale in Informatica.

Insegnamento**MFN1348 - Agenti Intelligenti**

Insegnamento (inglese):	Intelligent agents
CFU:	6
Settore:	INF/01 - INFORMATICA
Periodo didattico:	2
Tipologia di Attività Formativa:	B - caratterizzante
Docenti:	Alberto MARTELLI (Professore a Contratto)

1. Prerequisiti e Propedeuticità:**Competenze attese in ingresso**

Il corso assume le seguenti conoscenze:

- familiarità con le tecniche di programmazione sequenziale e concorrente in linguaggi di alto livello come Java, e capacità di comprendere e formulare algoritmi,
- conoscenza dei concetti di base dell'intelligenza artificiale come risoluzione automatica di problemi e rappresentazione della conoscenza,
- familiarità con i concetti di base della logica classica.

Eventuali corsi propedeutici

Le competenze del primo punto sono fornite dai corsi di programmazione e algoritmi della laurea triennale. Le competenze sugli altri due punti sono fornite dal corso di "Sistemi Intelligenti" della laurea triennale o, relativamente al terzo punto, da un corso di logica della laurea triennale. Gli studenti che non hanno seguito il corso di "Sistemi Intelligenti" si possono documentare consultando i capitoli iniziali di un testo di Intelligenza Artificiale (ad esempio S. Russell, P. Norvig. Intelligenza Artificiale, Un approccio moderno, terza edizione, Pearson/Prentice Hall).

2. Obiettivi formativi:

Il corso ha l'obiettivo di introdurre gli aspetti principali dei sistemi multiagente, ossia sistemi composti di elementi computazionali che interagiscono, noti come agenti. Gli agenti sono sistemi computazionali capaci di eseguire azioni in modo autonomo, e di interagire con altri agenti svolgendo attività sociali come cooperazione, coordinamento, negoziazione. I sistemi multi agente costituiscono una metafora naturale per modellare un ampio spettro di "artificial social systems". Il corso consiste di una parte metodologica e di una di laboratorio. Nella parte metodologica verranno trattate le architetture di singoli agenti e le principali problematiche legate all'interazione fra agenti. Nella parte di laboratorio saranno presentati alcuni linguaggi e ambienti per agenti, e sarà proposto qualche semplice sistema multiagente da implementare con questi strumenti.

3. Risultati dell'apprendimento attesi:

Al termine del corso lo studente dovrà aver acquisito le metodologie e gli strumenti formali per modellare sistemi software mediante un approccio ad agenti, descritti nella parte metodologica del corso. Lo studente dovrà inoltre essere in grado di implementare sistemi ad agenti mediante i linguaggi o gli ambienti presentati nella parte di laboratorio.

4. Modalità di verifica dell'apprendimento:

L'esame consiste in una interrogazione sugli argomenti trattati nella parte metodologica del corso e nella discussione e dimostrazione dei sistemi sviluppati nella parte di laboratorio.

5. Modalità d'insegnamento:

Le lezioni per la parte metodologica si svolgono in modo tradizionale con il supporto di slide e

lavagna. Nelle lezioni di laboratorio il docente presenterà linguaggi e ambienti per lo sviluppo di sistemi multiagente, interagendo con gli studenti nello sviluppo dei problemi proposti.

6. Attività di supporto:

Il materiale del corso sarà disponibile sul supporto on-line al corso I-learn.

7. Programma:

- Introduzione agli agenti intelligenti.
- Agenti BDI (Belief, Desire, Intention).
- Logiche per sistemi multiagente: logica modale, temporale, logiche per knowledge e belief.
- Linguaggi per agenti basati sulla logica.
- Agenti reattivi e ibridi.
- Multiagent Systems.
- Comunicazione fra agenti: Agent Communication Languages.
- Protocolli di interazione.
- Verifica di proprietà.
- Interazioni fra agenti: teoria dei giochi, strategie e equilibri di Nash.
- Come raggiungere un accordo: negoziazione.
- Linguaggi e ambienti per sistemi multiagente: Jade e Jason.

8. Testi consigliati e bibliografia:

Testo Principale:

Michael Wooldridge, An Introduction to MultiAgent Systems. John Wiley & Sons, Second Edition, 2009.

Altri:

- Fabio Luigi Bellifemine, Giovanni Caire, Dominic Greenwood, Developing Multi-Agent Systems with JADE (Wiley Series in Agent Technology), John Wiley & Sons, 2007.

- Rafael H. Bordini, Michael Wooldridge, Jomi Fred Hubner, Programming Multi-Agent Systems in Agentspeak Using Jason (Wiley Series in Agent Technology), John Wiley & Sons, 2007.

Insegnamento**MFN0997 - Algoritmi e Complessità - Parte A**

Insegnamento (inglese):	Algorithms and Complexity - A
CFU:	6
Settore:	INF/01 - INFORMATICA
Periodo didattico:	2
Tipologia di Attività Formativa:	B - caratterizzante
Docenti:	Maddalena ZACCHI (Titolare)

1. Prerequisiti e Propedeuticità:**Competenze attese in ingresso**

Lo studente deve conoscere i contenuti dei corsi di base di informatica e avere esperienza di programmazione. In particolare deve padroneggiare le più comuni tecniche di progettazione di algoritmi, come la tecnica "Divide et Impera" e la tecnica "Greedy", e avere familiarità con la nozione di complessità concreta per algoritmi iterativi e ricorsivi.

Eventuali corsi propedeutici

Le competenze richieste per una proficua frequenza del corso sono fornite dagli insegnamenti di Programmazione e Algoritmi della laurea in Informatica.

2. Obiettivi formativi:

Il corso ha lo scopo di fornire nozioni avanzate per il progetto, l'analisi ed il confronto di algoritmi. Un ulteriore obiettivo del corso è quello di far conoscere tecniche di programmazione che sono state studiate per affrontare problemi non trattabili, eventualmente fornendo soluzioni approssimate o risposte solo probabilmente corrette.

La preparazione di un seminario intende da una parte stimolare lo studente a cercare e selezionare documentazione sull'argomento proposto e dall'altra a porsi l'obiettivo di rendere fruibile l'argomento ai compagni attraverso una presentazione sufficientemente tecnica, anche se non troppo dettagliata, che evidenzi gli aspetti più interessanti.

3. Risultati dell'apprendimento attesi:

La tesi di Church-Turing sulla calcolabilità va rimeditata e rivista in riferimento ai costi effettivi che le computazioni richiedono. Lo studente dovrebbe essere in grado di valutare l'efficienza degli algoritmi e, qualora richiedano costi enormi in pratica, affrontare il problema di trovare programmi rapidi e poco costosi, che forniscano comunque soluzioni accettabili.

La presentazione del seminario ha come obiettivo quello di sviluppare la capacità di valutare e sintetizzare gli aspetti fondamentali di un argomento e la capacità di comunicarli.

4. Modalità di verifica dell'apprendimento:

Esame orale e presentazione di un seminario di approfondimento su uno degli argomenti del corso. L'esame orale prevede anche lo sviluppo di algoritmi di programmazione dinamica e di backtracking.

5. Modalità d'insegnamento:

Nel corso delle lezioni è previsto lo svolgimento di esercizi in aula per ognuna delle parti fondamentali in cui si divide l'insegnamento: programmazione dinamica, backtracking e branch-and-bound, algoritmi non deterministici e loro trasformazione in deterministici, classi di complessità, algoritmi approssimati.

Le lezioni finali dell'insegnamento vengono dedicate alla preparazione di un seminario da parte di studenti divisi in gruppi di due/tre persone.

Agli studenti sono resi disponibili i lucidi eventualmente usati dal docente come supporto alle lezioni.

6. Attività di supporto:

Il materiale didattico di supporto (programma dettagliato, lucidi, esempi di testi d'esame ed altro) è disponibile sul supporto on-line ai corsi I-learn.

7. Programma:

Progetto e analisi di Algoritmi - Programmazione dinamica. Uso "top-down" e "bottom-up" di definizioni ricorsive di funzioni. Esempi: Zaino, Cammini minimi nei grafi, Prodotto di matrici. - Backtracking. Esempi: Il problema delle n regine, Commesso Viaggiatore. - Branch and Bound. Esempi: Zaino, Commesso Viaggiatore.

Complessità polinomiale in tempo - Certificati polinomiali, algoritmi non deterministici, le classi P e NP - Riducibilità polinomiale, NP-completezza.

Algoritmi di approssimazione. - Approssimazione assoluta ed errore relativo. - Schemi di approssimazione pienamente polinomiali.

Algoritmi randomizzati e algoritmi euristici.

Il programma dettagliato del corso sarà pubblicato sul supporto on-line ai corsi I-learn

8. Testi consigliati e bibliografia:

T. H.Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, "Introduzione agli algoritmi e strutture dati", Seconda edizione, McGraw-Hill, 2005

A. Bertossi, "Algoritmi e strutture di dati", UTET Libreria, 2000.

C.H. Papadimitriou, "Computational Complexity", Addison-Wesley, Longman, 1995

R. Sedgewick, "Algoritmi in C", Addison-Wesley, Masson, 1993

Insegnamento**INF0007 - Analisi e Visualizzazione di Reti Complesse**

Insegnamento (inglese):	Analysis and Visualization of Complex Networks
CFU:	9
Settore:	INF/01 - INFORMATICA
Periodo didattico:	2
Tipologia di Attività Formativa:	B - caratterizzante
Docenti:	Giancarlo Francesco RUFFO (Titolare)

1. Prerequisiti e Propedeuticità:**Competenze attese in ingresso**

Una forte conoscenza operativa di nozioni di probabilità e di algebra lineare (a livello di laurea in disciplina scientifica) sarà senz'altro d'aiuto, così come una generale maturità in ambito matematico.

La capacità di scrivere codice senza problemi è importante, poiché abilità di programmazione sono richieste per eseguire il progetto finale del corso.

Eventuali corsi propedeutici

Se lo studente proviene da un nostro corso di laurea, deve aver sostenuto i seguenti esami:

MFN0570 - Analisi Matematica

MFN0588 - Calcolo Matriciale e Ricerca Operativa

MFN0600 - Elementi di Probabilità e Statistica

MFN0582 - Programmazione I

MFN0585 - Programmazione II

MFN0597 - Algoritmi e Strutture Dati

MFN0602 - Basi di Dati

MFN0598 - Fisica

MFN0634 - Tecnologie Web

MFN0608 - Interazione Uomo Macchina e Tecnologie Web

Altrimenti, se lo studente proviene da altri corsi di laurea, dovrà verificare se i corsi suddetti hanno un programma equivalente a quelli degli esami da loro superati.

2. Obiettivi formativi:

Questo corso introduce i concetti, i principi e le metodologie principali nel campo interdisciplinare della cosiddetta Scienza delle Reti, con un'attenzione particolare alle tecniche analitiche, alla modellazione e alle applicazioni per il Web e per i Social Media.

Gli argomenti trattati includono lo studio della struttura di una rete (grafo), i modelli matematici delle reti, le topologie delle reti più comuni, la struttura di grafi di grandi dimensioni, le strutture

delle comunità, la diffusione epidemica di virus ed informazioni, il PageRank e altre misure di centralità, processi dinamici nelle reti e visualizzazione di grafi.

Un ulteriore obiettivo di questo corso (che lo differenzia da MFN0954-"Reti Complesse") rientra nell'ambito della visualizzazione dati scientifici. Gli studenti apprenderanno i principi base della progettazione di un sistema di visualizzazione, apprendendo come acquisire, esplorare e analizzare insiemi di dati di grandi dimensioni. Gli studenti apprenderanno inoltre tecniche per visualizzare dati multivariati, temporali, testuali, geospaziali, gerarchici e (soprattutto) basati su relazioni di rete e/o grafi. Infine, gli studenti useranno strumenti quali Gephi, D3, R e ggplot2 (oltre a ricevere una breve introduzione a Processing) e tanti altri strumenti per prototipare molte di queste tecniche su insiemi di dati esistenti.

3. Risultati dell'apprendimento attesi:

Dopo aver superato con successo l'esame di questo corso gli studenti saranno in grado di:

- Definire e calcolare metriche di rete di base
- Descrivere le caratteristiche strutturali di reti tecno-sociali
- Mettere in relazione le proprietà del grafo con le funzioni e l'evoluzione della rete corrispondente
- Mettere in relazione proprietà locali con l'emergenza di schemi globali
- Esplorare nuove angolazioni per capire i comportamenti collettivi che si osservano in una rete
- Impostare e condurre analisi su dati di reti di grandi dimensioni
- Usare strumenti computazionali per l'analisi di rete, quali la libreria igraph (per R e Python) e Gephi.
- Visualizzare reti per evidenziarne proprietà strutturali e globali
- Mettere in relazione la visualizzazione dei dati al processo di analisi quantitativa vera e propria
- Impostare e progettare un sistema interattivo di visualizzazione di dati complessi all'interno di un processo di analisi dati.
- Usare strumenti di visualizzazione dati quali Gephi, open API basate su javascript (es., Google Chart, D3) e conoscere le basi di altri strumenti molto noti (es., Processing, R, ggplot2)

4. Modalità di verifica dell'apprendimento:

Esame orale (60%).

Compito I (20%): relazione scritta (2000-3000 parole).

Compito II (20%): progetti individuali su un'analisi di dati rappresentati in forma di rete (lo sviluppo di codice è normalmente parte del compito).

Per superare l'esame gli studenti devono raggiungere e superare un totale del 60% quando tutti le singoli parti sono state terminate e sommate insieme.

5. Modalità d'insegnamento:

Le lezioni si svolgeranno nella modalità tradizionale frontale, facendo uso sia di lavagna che di diapositive elettroniche, con sperimentazione interattiva degli strumenti computazionali di analisi descritti.

Sono previste testimonianze ed interventi di carattere seminariale di esperti e studiosi anche di altre discipline.

6. Attività di supporto:

Una pagina moodle è stata creata per il corso. Tutto il materiale, così come appunti, note e risorse on line saranno condivise. Usando la piattaforma Moodle, gli studenti saranno in grado di discutere le idee introdotte durante il corso e porre domande al docente e agli altri studenti.

7. Programma:

Scienza delle reti

- Intorduzione alle reti complesse
- Teoria dei grafi e metriche di rete
- Reti casuali
- Reti "Piccolo Mondo"
- Reti ad invarianza di scala
- Reti statiche ed in evoluzione
- Correlazioni di grado
- Comunità
- Fenomeni di diffusione
- Apprendimento e giochi su reti

Casi di studio ed applicazioni

- La struttura del nucleo di Internet - evoluzioni e modellazione
- Struttura del Web - PageRank e reti di documenti
- Reti sociali on line e Social Media - Twitter, Facebook, Amazon, ...
- Visualizzazione di Reti
- Reti di similarità e sistemi di raccomandazione
- Fenomeno del "ricco che si arricchisce"
- Collegamenti, vicini e comunità
- Fenomeni "a cascata" ed epidemici/virali
- Bilanciamento strutturale di una rete
- Analisi del sentimento e delle componenti spaziali e temporali di una rete sociale on line.

Visualizzazione Dati

- Grafici e plot di base, visualizzazione di dati multivariati
- Principi di percezione, colore, progettazione e valutazione
- Visualizzazione testuale
- Interattività e animazione
- Visualizzazione di dati temporali

- Visualizzazione di dati geospaziali
- Visualizzazione di dati gerarchici
- Visualizzazione di dati di rete

8. Testi consigliati e bibliografia:

Complex Networks:

A.-L. Barabási, Network Science, Cambridge University Press, 2015 (online version: <http://barabasi.com/networksciencebook/>)

D. Easley and J. Kleinberg. Networks, Crowds, and Markets: Reasoning About a Highly Connected World, Cambridge University Press, 2010.

M. E. J. Newman. Networks: An Introduction, Oxford University Press, 2010.

Scientific Python:

W. McKinney. Python for Data Analysis. O'Reilly.2012

Data Visualization:

S. Murray, Interactive Data Visualization for the Web, O'Reilly Media, 2012.

N. Yau. Data Points: Visualization that means something. Wiley, 2012.

E. R. Tufte. The Visual Display of Quantitative Information. Graphic Press, 2013

Insegnamento**MFN0943 - Apprendimento Automatico e Analisi Intelligente dei Dati**

Insegnamento (inglese):	Machine Learning and Intelligent Data Analysis
CFU:	9
Settore:	INF/01 - INFORMATICA
Periodo didattico:	0
Tipologia di Attività Formativa:	B - caratterizzante
Docenti:	Roberto ESPOSITO (Titolare) Rosa MEO (Titolare)

1. Prerequisiti e Propedeuticità:**Competenze attese in ingresso**

Conoscenze elementari di probabilità e statistica, algoritmi, sistemi informativi e basi di dati.

Eventuali corsi propedeutici

Sistemi intelligenti (laurea triennale), Basi di dati (laurea triennale: corso di Basi dati e Sperimentazioni).

2. Obiettivi formativi:

Gli obiettivi del corso introdurranno gli studenti al campo dell'Apprendimento Automatico, che unisce competenze di informatica (intelligenza artificiale, algoritmi, ottimizzazione, basi di dati) e statistica.

Il corso insegna le differenze tra problemi e modelli e introduce gli studenti ad alcuni modelli popolari nell'Apprendimento Automatico, tra cui la classificazione binaria, la trasformazione di una classificazione binaria ad una multi-classe, l'apprendimento di concetti tramite formule logiche, i modelli ad albero, i modelli a regole, la ricerca di sottogruppi (subgroup discovery), i modelli lineari (dei minimi quadrati, la regressione), il perceptrone, le Support Vector Machines, i metodi Kernel. Si discute la decomposizione bias-varianza e il problema dell'*overfitting*. Tratterà i modelli basati su distanza, come i k-nearest neighbors, il clustering basato sulle k-medie, il clustering gerarchico, e basato su densità. Inoltre tratterà la distinzione tra i tipi di attributi (feature), le operazioni ammesse e le descrizioni statistiche, la trasformazione tra le features (normalizzazione, discretizzazione, calibrazione, ecc). Infine tratterà l'apprendimento di modelli probabilistici come il metodo della massima verosimiglianza, la regressione logistica, i modelli Bayesiani e naive Bayes ed Expectation-Maximization. L'ultima parte del corso tratta delle misure di validazione dei modelli e della verifica di significatività statistica dei risultati. La parte di laboratorio del corso introdurrà gli studenti a una suite software open source (Weka) che include gli algoritmi di apprendimento dei modelli visti durante il corso (e molto altro). Con Weka si svilupperanno sessioni di analisi dei dati usando dataset pubblicamente disponibili.

3. Risultati dell'apprendimento attesi:

1. Conoscenza dei principali modelli di apprendimento automatico
2. Conoscenze di statistica di base per l'analisi dei dati, quali test di significatività statistica, test di indipendenza
3. Acquisizione delle modalità in cui avviene il processo di analisi dei dati, e relativa scelta delle tecniche da adottare in ciascuna fase del processo.
4. Scelta della tecnica di analisi da adottare per un certo insieme di dati
5. Analisi di un algoritmo di apprendimento automatico
6. Interpretazione e validazione dei risultati dell'analisi

4. Modalità di verifica dell'apprendimento:

Colloquio orale in cui si verifica l'apprendimento dei contenuti teorici del corso e la sperimentazione sui sistemi e gli algoritmi. Qui sono elencate a titolo d'esempio alcune domande che gli anni scorsi si facevano durante il colloquio.

1. Descrivere le principali operazioni di trasformazione tra le features
2. Illustrare gli obiettivi dell'apprendimento dei modelli ad albero
3. Illustrare la differenza tra i modelli a lista e a insieme di regole
4. Scrivere lo pseudo-codice di alcuni algoritmi di classificazione o clustering o di estrazione di pattern frequenti
5. Illustrare l'algoritmo di Expectation-Maximization
6. Descrivere le misure e tecniche di validazione del processo di classificazione.
7. Descrivere il significato e l'utilizzo della statistica del Chi-square.
8. Illustrare il significato intuitivo di una regola di associazione.
9. Descrivere lo spazio di ricerca dei large itemset.
10. Descrivere l'algoritmo e il principio di Apriori.
11. Spiegare il significato di *closed* itemset e i suoi vantaggi
12. Descrivere e confrontare alcuni degli approcci per fare clustering
13. Illustrare uno degli esercizi svolti in laboratorio con Weka.

5. Modalità d'insegnamento:

Le lezioni in aula sono svolte principalmente con l'ausilio del calcolatore (proiezione di lucidi animati). Le esercitazioni in Laboratorio saranno svolte ad un calcolatore su cui è stato installato il software da sperimentare (Weka) e sul cui utilizzo lo studente verrà anche valutato in sede di esame.

6. Attività di supporto:

1. Il materiale didattico di supporto (lucidi, link, esercizi ed altro) è disponibile presso il supporto on-line ai corsi [Learn](#).
2. Slides del libro "Peter Flach, Machine Learning - The Art and Science of Algorithms that Make Sense of Data, Cambridge University Press, 2012" aggiornate dal docente.
3. [WEKA](#)
4. [UCI KDD Archive](#) (repository di data set per vari task di analisi dei dati).

7. Programma:

1. Introduzione
2. differenza tra problema e modello
3. la classificazione binaria e la trasformazione di una classificazione binaria ad una multi-classe
4. l'apprendimento di concetti tramite formule logiche
5. i modelli ad albero, l'apprendimento e i loro obiettivi
6. i modelli a regole,
7. la ricerca di sottogruppi (subgroup discovery)
8. i modelli lineari (dei minimi quadrati, la regressione), il perceptrone,
9. le Support Vector Machines e i metodi Kernel.
10. decomposizione bias-varianza
11. *overfitting*.
12. modelli basati su distanza (k-nearest neighbors, clustering k-medie, clustering gerarchico, clustering basato su densità).
13. tipi di attributi (feature), operazioni ammesse e descrizioni statistiche
14. trasformazione tra le features (normalizzazione, discretizzazione, calibrazione, ecc).
15. modelli probabilistici (massima verosimiglianza, regressione logistica, modelli Bayesiani e naive Bayes, Expectation-Maximization).
16. Validazione dei modelli e verifica di significatività statistica dei risultati.
17. Sperimentazione con un sistema reale di analisi dei dati su vari data set: Weka

Nota: L'ordine degli argomenti è indicativo.

8. Testi consigliati e bibliografia:

* Peter Flach, Machine Learning - The Art and Science of Algorithms that Make Sense of Data, Cambridge University Press, 2012.

Libri di consultazione:

* Data Mining - Practical Machine Learning Tools and Techniques with Java Implementations, (2nd edition), Ian H. Witten, Eibe Frank, Morgan Kaufmann, 2006.

* Probability and Statistics for Engineering and the Science Jay L. Devore, Duxbury, 2004, 6th edition.

Insegnamento**INF0010 - Apprendimento Automatico e Analisi Intelligente dei Dati - Parte A**

Insegnamento (inglese):	Machine Learning and Intelligent Data Analysis - A
CFU:	6
Settore:	INF/01 - INFORMATICA
Periodo didattico:	0
Tipologia di Attività Formativa:	B - caratterizzante
Docenti:	Roberto ESPOSITO (Titolare) Rosa MEO (Titolare)

1. Prerequisiti e Propedeuticità:**Competenze attese in ingresso**

Conoscenze elementari di probabilità e statistica, algoritmi, sistemi informativi e basi di dati.

Eventuali corsi propedeutici

Sistemi intelligenti (laurea triennale), Basi di dati (laurea triennale: corso di Basi dati e Sperimentazioni).

2. Obiettivi formativi:

Gli obiettivi del corso introdurranno gli studenti al campo dell'Apprendimento Automatico, che unisce competenze di informatica (intelligenza artificiale, algoritmi, ottimizzazione, basi di dati) e statistica. Il corso insegna le differenze tra problemi e modelli e introduce gli studenti ad alcuni modelli popolari nell'Apprendimento Automatico. Il programma del corso consiste di una parte introduttiva (le differenze tra problemi e modelli) seguita dalla discussione di una serie di modelli. I docenti concorderanno con gli studenti quali degli argomenti che seguono inserire nel proprio piano di studi e seguiranno le lezioni relative a quegli argomenti.

La classificazione (binaria e multi-classe), l'apprendimento di concetti tramite formule logiche, i modelli ad albero, i modelli a regole, la ricerca di sottogruppi (subgroup discovery), i modelli lineari (dei minimi quadrati, la regressione), il perceptrone, le Support Vector Machines, i metodi Kernel. Si discute la decomposizione bias-varianza e il problema dell'*overfitting*. Tratterà i modelli basati su distanza, come i k-nearest neighbors, il clustering basato sulle k-medie, il clustering gerarchico, e basato su densità. Inoltre tratterà la distinzione tra i tipi di attributi (feature), le operazioni ammesse e le descrizioni statistiche, la trasformazione tra le features (normalizzazione, discretizzazione, calibrazione, ecc). Infine tratterà l'apprendimento di modelli probabilistici come il metodo della massima verosimiglianza, la regressione logistica, i modelli Bayesiani e naive Bayes ed Expectation-Maximization. L'ultima parte del corso tratta delle misure di validazione dei modelli e della verifica di significatività statistica dei risultati. La parte di laboratorio del corso introdurrà gli studenti a una suite software open source (Weka) che include gli algoritmi di apprendimento dei modelli visti durante il corso (e molto altro). Con Weka si svilupperanno sessioni di analisi dei dati usando dataset pubblicamente disponibili.

3. Risultati dell'apprendimento attesi:

1. Conoscenza dei principali modelli di apprendimento automatico
2. Conoscenze di statistica di base per l'analisi dei dati, quali test di significatività statistica, test di indipendenza
3. Acquisizione delle modalità in cui avviene il processo di analisi dei dati, e relativa scelta delle tecniche da adottare in ciascuna fase del processo.
4. Scelta della tecnica di analisi da adottare per un certo insieme di dati
5. Analisi di un algoritmo di apprendimento automatico
6. Interpretazione e validazione dei risultati dell'analisi

4. Modalità di verifica dell'apprendimento:

Colloquio orale in cui si verifica l'apprendimento dei contenuti teorici del corso e la sperimentazione sui sistemi e gli algoritmi. Qui sono elencate a titolo d'esempio alcune domande che gli anni scorsi si facevano durante il colloquio.

1. Descrivere le principali operazioni di trasformazione tra le features
2. Illustrare gli obiettivi dell'apprendimento dei modelli ad albero
3. Illustrare la differenza tra i modelli a lista e a insieme di regole
4. Scrivere lo pseudo-codice di alcuni algoritmi di classificazione o clustering o di estrazione di pattern frequenti
5. Illustrare l'algoritmo di Expectation-Maximization
6. Descrivere le misure e tecniche di validazione del processo di classificazione.
7. Descrivere il significato e l'utilizzo della statistica del Chi-square.
8. Illustrare il significato intuitivo di una regola di associazione.
9. Descrivere lo spazio di ricerca dei large itemset.
10. Descrivere l'algoritmo e il principio di Apriori.
11. Spiegare il significato di *closed* itemset e i suoi vantaggi
12. Descrivere e confrontare alcuni degli approcci per fare clustering
13. Illustrare uno degli esercizi svolti in laboratorio con Weka.

5. Modalità d'insegnamento:

Le lezioni in aula sono svolte principalmente con l'ausilio del calcolatore (proiezione di lucidi animati). Le esercitazioni in Laboratorio saranno svolte ad un calcolatore su cui è stato installato il software da sperimentare (Weka) e sul cui utilizzo lo studente verrà anche valutato in sede di esame.

6. Attività di supporto:

1. Il materiale didattico di supporto (lucidi, link, esercizi ed altro) è disponibile presso il supporto on-line ai corsi [I-learn](#).
2. Slides del libro "Peter Flach, Machine Learning - The Art and Science of Algorithms that Make Sense of Data, Cambridge University Press, 2012" aggiornate dal docente.
3. [WEKA](#)
4. [UCI KDD Archive](#) (repository di data set per vari task di analisi dei dati).

7. Programma:

1. Introduzione
2. differenza tra problema e modello
3. la classificazione binaria e la trasformazione di una classificazione binaria ad una multi-classe
4. l'apprendimento di concetti tramite formule logiche
5. i modelli ad albero, l'apprendimento e i loro obiettivi
6. i modelli a regole,
7. la ricerca di sottogruppi (subgroup discovery)
8. i modelli lineari (dei minimi quadrati, la regressione), il perceptrone,
9. le Support Vector Machines e i metodi Kernel.
10. decomposizione bias-varianza
11. *overfitting*.
12. modelli basati su distanza (k-nearest neighbors, clustering k-medie, clustering gerarchico, clustering basato su densità).
13. tipi di attributi (feature), operazioni ammesse e descrizioni statistiche
14. trasformazione tra le features (normalizzazione, discretizzazione, calibrazione, ecc).
15. modelli probabilistici (massima verosimiglianza, regressione logistica, modelli Bayesiani e naive Bayes, Expectation-Maximization).
16. Validazione dei modelli e verifica di significatività statistica dei risultati.
17. Sperimentazione con un sistema reale di analisi dei dati su vari data set: Weka

Nota: L'ordine degli argomenti è indicativo.

8. Testi consigliati e bibliografia:

* Peter Flach, Machine Learning - The Art and Science of Algorithms that Make Sense of Data, Cambridge University Press, 2012.

Libri di consultazione:

* Data Mining - Practical Machine Learning Tools and Techniques with Java Implementations, (2nd edition), Ian H. Witten, Eibe Frank, Morgan Kaufmann, 2006.

* Probability and Statistics for Engineering and the Science Jay L. Devore, Duxbury, 2004, 6th edition.

Insegnamento**MFN0969 - Architettura degli Elaboratori II**

Insegnamento (inglese):	Computer Architecture II
CFU:	6
Settore:	INF/01 - INFORMATICA
Periodo didattico:	1
Tipologia di Attività Formativa:	B - caratterizzante
Docenti:	Marco ALDINUCCI (Titolare) Daniele GUNETTI (Titolare)

1. Prerequisiti e Propedeuticità:**Competenze attese in ingresso**

Agli studenti è richiesta una conoscenza di base dell'architettura di un computer (secondo quanto studiato nel corso di Architetture degli Elaboratori I) e dei concetti di base dei sistemi operativi (secondo quanto studiato nel corso di Sistemi operativi).

Eventuali corsi propedeutici

Architetture degli elaboratori I e Sistemi Operativi

2. Obiettivi formativi:

La conoscenza approfondita dell'architettura interna dei moderni computer è un requisito fondamentale del curriculum di qualsiasi professionista dell'informatica. Conoscere le diverse tipologie di processori offerti dal mercato permette di scegliere quelli più adeguati per il tipo di problema che si vuole risolvere e del tipo di servizio che si vuole fornire, raggiungendo il giusto compromesso tra costi e prestazioni.

NOTA IMPORTANTE: TUTTE LE INFORMAZIONI SUL CORSO, E IL MATERIALE DIDATTICO SI TROVANO ALLA PAGINA DEL CORSO:

<http://www.di.unito.it/~gunetti/DIDATTICA/architettureII/index.html>

3. Risultati dell'apprendimento attesi:

Lo studente acquisirà la conoscenza dettagliata dell'architettura e del funzionamento dei moderni processori, e dovrà essere in grado di ragionare qualitativamente sulle prestazioni che ogni tipologia di processori è in grado di fornire in base al tipo di problema e algoritmo adottato. Avrà inoltre appreso i fondamenti della programmazione assembler e delle moderne applicazioni basate su GPGPU.

4. Modalità di verifica dell'apprendimento:

Esame orale, nel quale si verificherà la capacità dello studente di ragionare e di mettere in relazione fra loro gli argomenti del corso. Durante il corso domande, esercizi e discussioni con gli studenti su temi scelti.

5. Modalità d'insegnamento:

Sono previste 60 ore di lezione frontali che seguono il programma riportato più avanti, integrate da casi di studio di processori e architetture reali.

6. Attività di supporto:

Lucidi disponibili sulla pagina dei corsi del docente

7. Programma:

* PARTE I: o Concetti di base delle architetture RISC

o Concetti di base del Pipelining

o Instruction Level Parallelism

o Instruction Level Parallelism

o Concetti fondamentali di Caching

* PARTE II: o Introduzione -- Multithreading

o Architetture Multiprocessore

o Architetture Multicomputer ~~Processori vettoriali~~ una nota storica

MISCELLANEA: o Sistemi RAID o Cenni sulla programmazione in assembler dell'8088 o seminario sulle architetture grafiche

8. Testi consigliati e bibliografia:

* Andrew Tanenbaum: Structured Computer Organization (fifth Edition)

- D. Patterson & J. Hennessy: Struttura e Progetto dei Calcolatori. L'interfaccia Hardware-Software (2a ed. - terza ed. americana)
- J. Hennessy & D. Patterson: Computer Architecture, A quantitative Approach (3rd Edition)
- J. Hennessy & D. Patterson: Computer Architecture, A quantitative Approach (4th Edition)
- J. Hennessy & D. Patterson: Computer Architecture, A quantitative Approach (5th Edition)
- Agli studenti sono resi disponibili fin dall'inizio del corso i lucidi usati dal docente a lezione

Insegnamento**MFN0947 - Basi di Dati Multimediali**

Insegnamento (inglese):	Multimedia Databases
CFU:	9
Settore:	INF/01 - INFORMATICA
Periodo didattico:	2
Tipologia di Attività Formativa:	B - caratterizzante
Docenti:	Maria Luisa SAPINO (Titolare)

1. Prerequisiti e Propedeuticità:**Competenze attese in ingresso**

Conoscenze di base sulle basi di dati relazionali e ad oggetti.

Eventuali corsi propedeutici

Propedeutico: Basi di Dati

Consigliato: Modelli Avanzati e Architetture di Basi di Dati

2. Obiettivi formativi:

Il corso illustra gli aspetti che caratterizzano le basi di dati multimediali, in particolare confrontandone le caratteristiche con quelle delle basi di dati standard. L'obiettivo primario è quello di presentare i problemi principali (e le corrispondenti soluzioni) nella rappresentazione, memorizzazione, e nel retrieval di dati multimediali, in particolare testi, immagini e video, e più in generale nella gestione di grosse moli di dati eterogenei (big data), spesso imprecisi e/o incompleti.

3. Risultati dell'apprendimento attesi:

Conoscenza dei problemi (e di alcune loro soluzioni) relativi alla archiviazione ed al reperimento di informazioni multimediali, dalla rappresentazione, all'indicizzazione, agli algoritmi di interrogazione con ranking dei risultati.

Gestione di dati eterogenei complessi.

4. Modalità di verifica dell'apprendimento:

L'esame è orale, e prevede domande di carattere generale, che consentano al docente di valutare quanto lo studente padroneggi la visione di insieme dei problemi trattati, oltre ad alla trattazione puntuale di temi specifici affrontati (ed eventualmente sperimentati) durante il corso, ad esempio un metodo di indicizzazione, di clustering o di relevance feedback. Durante l'esame orale viene anche discusso e valutato il software sviluppato nella fase di approfondimento sperimentale del corso.

5. Modalità d'insegnamento:

Le lezioni si svolgono in aula, con la proiezione di diapositive a cura del docente. Per alcuni argomenti svolti vengono proposte attività di sperimentazione.

In particolare, 6 crediti sono dedicati alle lezioni di carattere teorico di presentazione di contenuti, e 3 crediti sono dedicati ad attività di approfondimento, anche sperimentale.

6. Attività di supporto:

Copia dei lucidi viene regolarmente messa a disposizione degli studenti mediante la piattaforma Moodle

7. Programma:

Caratteristiche dei dati multimediali, e requisiti che un sistema di gestione di dati multimediali deve soddisfare.

Scelta delle features nella rappresentazione di dati multimediali.

Il modello vettoriale per la rappresentazione di dati multimediali.

Indicizzazione in spazi multidimensionali.

Clustering di dati multimediali

Strategie per la valutazione efficiente di query multimediali.

Relevance feedback

Gestione di dati complessi, quali le reti sociali.

Durante il corso vengono inoltre organizzati seminari su temi specifici, a cura di colleghi di universita' straniere e di ricercatori presso centri di ricerca dell' area torinese.

8. Testi consigliati e bibliografia:

Libro di testo: K. Selcuk Candan, M.L. Sapino. Data Management for Multimedia Retrieval. Cambridge University Press, 2010.

Insegnamento**MFN0993 - Basi di Dati Multimediali - Parte A**

Insegnamento (inglese):	Multimedia Databases - A
CFU:	6
Settore:	INF/01 - INFORMATICA
Periodo didattico:	2
Tipologia di Attività Formativa:	B - caratterizzante
Docenti:	Maria Luisa SAPINO (Titolare)

1. Prerequisiti e Propedeuticità:**Competenze attese in ingresso**

Conoscenze di base sulle basi di dati relazionali e ad oggetti.

Eventuali corsi propedeutici

Propedeutico: Basi di Dati

Consigliato: Modelli Avanzati e Architetture di Basi di Dati

2. Obiettivi formativi:

Il corso illustra gli aspetti che caratterizzano le basi di dati multimediali, in particolare confrontandone le caratteristiche con quelle delle basi di dati standard. L'obiettivo principale e' quello di presentare i problemi principali (e le corrispondenti soluzioni) nella rappresentazione, memorizzazione, e nel retrieval di testi, suoni, e immagini, e piu' in generale nella gestione di grosse moli di dati eterogenei, spesso imprecisi.

3. Risultati dell'apprendimento attesi:

Conoscenza dei problemi (e di alcune loro soluzioni) relativi alla archiviazione ed al reperimento di informazioni multimediali, dalla rappresentazione, all' indicizzazione, agli algoritmi di interrogazione con ranking dei risultati. Gestione di dati eterogenei complessi.

4. Modalità di verifica dell'apprendimento:

L'esame e' orale, e prevede domande di carattere generale, che consentano al docente di valutare quanto lo studente padroneggi la visione di insieme dei problemi trattati, oltre ad alla trattazione puntuale di temi specifici affrontati (ed eventualmente sperimentati) durante il corso, ad esempio un metodo di indicizzazione, o di clustering.

5. Modalità d'insegnamento:

Le lezioni si svolgono in aula, con la proiezione di diapositive a cura del docente.

6. Attività di supporto:

Copia dei lucidi viene regolarmente messa a disposizione degli studenti mediante la piattaforma Moodle.

7. Programma:

Il corso e' mutuato dall'analogo corso da 9 crediti, con cui condivide tutto il programma tranne i tre crediti di approfondimento sperimentale.

Caratteristiche dei dati multimediali, e requisiti che un sistema di gestione di dati multimediali deve soddisfare.

Scelta delle features nella rappresentazione di dati multimediali.

Il modello vettoriale per la rappresentazione di dati multimediali.

Indicizzazione in spazi multidimensionali.

Clustering di dati multimediali

Strategie per la valutazione efficiente di query multimediali.

Gestione di dati complessi, quali le reti sociali.

8. Testi consigliati e bibliografia:

K. Selcuk Candan, M.L. Sapino. Data Management for Multimedia Retrieval Cambridge University Press, 2010.

Insegnamento**MFN0951 - Bioinformatica**

Insegnamento (inglese):	Bioinformatics
CFU:	6
Settore:	INF/01 - INFORMATICA
Periodo didattico:	1
Tipologia di Attività Formativa:	B - caratterizzante
Docenti:	Marco BOTTA (Titolare) Raffaele Adolfo CALOGERO (Titolare)

1. Prerequisiti e Propedeuticità:**Competenze attese in ingresso**

Il modulo presuppone una conoscenza di base delle tematiche di algoritmi e della loro complessità, conoscenza delle tecniche di risoluzione dei problemi, basi di statistica e calcolo delle probabilità. Stati

Eventuali corsi propedeutici

Algoritmi e Strutture Dati. Sistemi Intelligenti Elementi di Probabilità e Statistica

2. Obiettivi formativi:

Introduzione ai problemi principali dell'analisi di dati biologici e descrizione dei relativi algoritmi utilizzati in ambito bioinformatico.

3. Risultati dell'apprendimento attesi:

Lo studente sarà in grado di individuare la tecnica più adatta da utilizzare per affrontare un problema di analisi di dati biologici. Inoltre, conoscendone il funzionamento, lo studente sarà anche in grado di proporre miglioramenti agli algoritmi esistenti.

4. Modalità di verifica dell'apprendimento:

L'esame consiste in una prova orale durante la quale lo studente deve dimostrare di avere acquisito le nozioni presentate durante le lezioni, rispondendo ad almeno tre domande sugli argomenti del programma svolto e presentare una breve relazione sul contenuto di un articolo scientifico concordato con il docente.

5. Modalità d'insegnamento:

Le lezioni saranno svolte principalmente in aula con ausilio di lucidi proiettati su schermo. Il materiale utilizzato a lezione è disponibile on-line sul sito della didattica.

6. Attività di supporto:

Per il materiale didattico on-line seguire i link alla voce moduli del corso.

7. Programma:

Introduzione alla biologia molecolare Introduzione alla biologia computazionale Panoramica dei problemi aperti L'allineamento di sequenze L'allineamento multiplo di sequenze Algoritmi di predizione e classificazione Algoritmi di clustering per la costruzione di alberi filogenetici Algoritmi di analisi di dati di microarray Next Generation Sequencing: problemi e algoritmi Gene Networks: algoritmi per la costruzione e l'inferenza Modelli per l'analisi di pathway metabolici

8. Testi consigliati e bibliografia:

Bioinformatics Algorithms: An Active Learning Approach, Phillip Compeau and Pavel Pevzner, Active Learning Publishers, 2014 An Introduction to Bioinformatics Algorithms (Computational Molecular Biology), Neil C. Jones and Pavel A. Pevzner, The MIT Press, 2004 Agli studenti sono anche resi disponibili fin dall'inizio del corso i lucidi usati dal docente nelle lezioni in aula.

Insegnamento**MFN0939 - Calcolabilità e Complessità B**

Insegnamento (inglese):	Computability and Complexity - B
CFU:	6
Settore:	INF/01 - INFORMATICA
Periodo didattico:	1
Tipologia di Attività Formativa:	B - caratterizzante
Docenti:	Simonetta RONCHI DELLA ROCCA (Titolare)

1. Prerequisiti e Propedeuticità:**Competenze attese in ingresso**

Sono richieste buone conoscenze di logica, programmazione e di algoritmi; inoltre si assume che lo studente posseda le nozioni di linguaggio formale, grammatica e di automa.

Eventuali corsi propedeutici

Matematica Discreta e Logica, Programmazione 1 e 2, Algoritmi e Strutture Dati, Linguaggi Formali e Traduttori.

2. Obiettivi formativi:

Che cos'è un algoritmo? Quali problemi si possono risolvere con un algoritmo? E in quali casi un algoritmo richiede risorse inaccessibili nella pratica? Il corso affronta questi problemi, trattando anzitutto la teoria della computabilità sia dal punto di vista classico - macchine di Turing, funzioni ricorsive - che da prospettive più attuali, come quella dei programmi while, legate ai linguaggi di programmazione. Si discutono poi i vari possibili criteri di misura delle risorse disponibili (tempo, memoria, cpu) e le classi di complessità, con particolare attenzione alla classica questione $P = NP$.

3. Risultati dell'apprendimento attesi:

Conoscenza delle basi teoriche dell'informatica, una maggiore consapevolezza delle limitazioni intrinseche all'uso delle macchine ed un'idea delle strategie per sopperire a tali limitazioni.

4. Modalità di verifica dell'apprendimento:

Esame orale. Verrà anche tenuto conto della partecipazione degli studenti agli esercizi in classe.

5. Modalità d'insegnamento:

Le lezioni si svolgono in aula, ed utilizzano la piattaforma Moodle sia per la distribuzione di ulteriore materiale didattico (come note e lucidi a cura del docente) che per le verifiche dell'apprendimento durante il corso.

6. Attività di supporto:

Vedi il sito moodle del corso

7. Programma:

Teoria della computabilità'

- Le Macchine di Turing

- Problemi non risolubili

- Funzioni ricorsive
- Calcolabilità e Linguaggi di Programmazione

Teoria della complessità

- Misure e classi di Complessità
- Classi di Complessità Temporale
- Classi di Complessità Spaziale
- Le classi P ed NP
- Problemi NP completi

8. Testi consigliati e bibliografia:

- C. Toffalori et alii, Teoria della calcolabilità e della complessità, McGraw-Hill 2005

- M. Sipser: "Introduction to the theory of Computation", Course Technology Ptr., 3^a edition, 2012.

Insegnamento**MFN0971 - Complementi di Analisi e Probabilità**

Insegnamento (inglese):	Elements of Analysis and Probability
CFU:	6
Settore:	MAT/05 - ANALISI MATEMATICA MAT/06 - PROBABILITA' E STATISTICA MATEMATICA
Periodo didattico:	1
Tipologia di Attività Formativa:	C - affine e integrativa
Docenti:	Enrico BIBBONA (Titolare) Paolo BOGGIATTO (Titolare)

1. Prerequisiti e Propedeuticità:**Competenze attese in ingresso**

Calcolo delle probabilità: eventi, indipendenza, probabilità congiunta e condizionata, variabili aleatorie discrete e continue e relativa caratterizzazione (distribuzione, funzione generatrice e momenti), teorema del limite centrale e legge dei grandi numeri; Analisi matematica: funzioni e relativo studio; calcolo differenziale ed integrale.

Eventuali corsi propedeutici

Calcolo delle Probabilità e Statistica (LT) Analisi Matematica 1 e 2 (LT)

2. Obiettivi formativi:

Il corso fornisce le competenze minimali per quanti avessero necessità di utilizzare il calcolo delle probabilità o le trasformate di Fourier per applicazioni di tipo modellistico, non pretende invece di permettere allo studente di operare autonomamente su tali tematiche. Obiettivo principale è infatti mettere lo studente nelle condizioni di proseguire eventualmente lo studio in modo autonomo, avendo acquisito il linguaggio necessario su alcuni temi avanzati su cui avrà le competenze sufficienti per poter leggere articoli scientifici che utilizzino tali strumenti a fini applicativi.

3. Risultati dell'apprendimento attesi:

Capacità di formalizzare realtà applicative con opportuni modelli e di studiarli con alcuni mezzi del calcolo delle probabilità o dell'analisi. Capacità di leggere articoli che facciano uso dei concetti introdotti nel corso e di risolvere semplici esercizi utilizzando le proprietà dei processi studiati.

4. Modalità di verifica dell'apprendimento:

Scritto e orale nel medesimo giorno: lo studente dovrà risolvere alcuni problemi scritti prima di sostenere l'orale.

5. Modalità d'insegnamento:

Lezioni frontali alternate con esercizi (svolti con il docente); gli studenti hanno la facoltà di consegnare alcuni esercizi assegnati e richiederne la correzione.

6. Attività di supporto:

Possono essere di supporto le note del Prof. Benenti alla pagina:
<http://www2.dm.unibo.it/~benenti/>

7. Programma:

variabili aleatorie congiuntamente distribuite; attese e probabilità condizionate; processi stocastici; catene di Markov a tempo discreto; processo di Poisson; catene di Markov a tempo continuo; Segnali e sistemi, funzione di trasferimento, filtri, modello di filtro passa-basso. Principali proprietà di serie e trasformata di Fourier e loro utilizzo in alcuni problemi di interesse applicativo. Cenni a rappresentazioni tempo-frequenza di segnali.

8. Testi consigliati e bibliografia:

S. Ross Introduction to Probability Models Academic Press Gasquet-Witomsky, Fourier Analysis and Applications, Springer

Insegnamento**INF0039 - Complementi di Reti e Sicurezza**

Insegnamento (inglese):	Advanced Network and Security
CFU:	12
Settore:	INF/01 - INFORMATICA
Periodo didattico:	1
Tipologia di Attività Formativa:	B - caratterizzante
Docenti:	Francesco BERGADANO (Titolare) Michele GARETTO (Titolare)

1. Prerequisiti e Propedeuticità:**Competenze attese in ingresso**

La prima parte del corso assume conoscenze di base sulle reti di calcolatori come commutazione di pacchetto/circuito, il controllo di errore, di flusso, di congestione, la pila protocollare, indirizzi e instradamento, Ethernet, l'architettura TCP/IP di Internet. Sono inoltre richieste nozioni elementari di analisi matematica, calcolo delle probabilità, processi stocastici.

Si presuppone la conoscenza dei sistemi operativi e di piattaforme di sviluppo e programmazione, quali Java e C++, considerata di aiuto alla comprensione degli argomenti svolti nel corso.

Eventuali corsi propedeutici

Reti di Calcolatori Complementi di analisi e probabilità

Sistemi Operativi

Sicurezza I (triennale)

2. Obiettivi formativi:

Negli ultimi due decenni, Internet è passata da strumento di ricerca a una componente fondamentale della società; qualcosa che noi tutti diamo per scontato e usiamo quotidianamente. In questo corso esploreremo perché l'infrastruttura di Internet è stata progettata in questo modo, i suoi principi di base e le scelte architetture. Esamineremo i pro e i contro della architettura attuale, e rifletteremo su come rendere Internet migliore in futuro.

Gli obiettivi del corso sono:

- Acquisire familiarità con lo stato dell'arte nelle reti di calcolatori: architetture, protocolli e sistemi.
- Ottenere una certa pratica nel leggere articoli di ricerca e comprenderli criticamente.
- Imparare a presentare e discutere efficacemente in pubblico un

argomento di reti.

La seconda parte del corso si propone di fornire agli studenti gli strumenti crittografici e tecnici utilizzati per garantire la sicurezza di reti e calcolatori. Inoltre, attraverso l'uso di esempi pratici, il corso fornisce agli studenti una comprensione concreta dei maggiori rischi di sicurezza e delle soluzioni disponibili. In particolare si farà riferimento alle tecnologie IAM (identity and access management).

3. Risultati dell'apprendimento attesi:

Per la prima parte, nozioni teoriche e metodologiche sul funzionamento e sulla progettazione di reti di calcolatori e di sistemi telematici complessi. Concetti avanzati sulla qualità del servizio.

Modelli di base per la simulazione e lo studio di reti di calcolatori. Principali metodologie per il controllo del traffico nelle reti a commutazione di pacchetto. Le principali tecnologie hardware e software ad oggi in uso nelle reti geografiche sia fisse che mobili. Metodologie e tecnologie per l'Ingegneria del Traffico; Caratteristiche avanzate dell'architettura TCP/IP e di Internet con particolare riferimento agli aspetti di routing e di offerta di servizi a qualità del servizio.

La seconda parte del corso si propone di preparare gli studenti a lavorare in azienda per la gestione dei sistemi informatici, cooperando con i livelli organizzativi per garantire la sicurezza del sistema informativo nel suo insieme. Questa è la naturale prosecuzione di quello di "Sicurezza" presente nel triennio. Le differenze tra i due corsi sono sostanziali: il corso della triennale fornisce il background teorico e strutturale sulle vulnerabilità dei computer e delle reti e sulle soluzioni derivanti dal campo della crittografia applicata. Lo scopo di questa seconda parte del corso è fornire le basi e gli strumenti necessari per affrontare problemi in un contesto reale, dove la gestione dei sistemi e delle reti deve tenere conto di policy interne e di questioni legate all'interoperabilità tra i diversi ambienti. Il corso comprende un laboratorio: è prevista l'emulazione di ambienti reali e la creazione di testbed nei quali inserire la definizione di utenti e la creazione di certificati di chiave pubblica.

4. Modalità di verifica dell'apprendimento:

Per la prima parte, la modalità di esame preferenziale (per il primo appello) consiste in:

- prova scritta su nozioni fondamentali del corso, nella forma di 5-6 domande a risposta aperta
- presentazione orale a fine corso (tramite slides) su un tema coordinato col docente

Gli appelli successivi prevedono una prova scritta analoga alla precedente e una prova orale.

Per la seconda parte del corso la verifica viene fatta sulle conoscenze acquisite durante il corso. La durata della prova scritta è di un'ora e prevede domande aperte e domande a risposta multipla, relative agli argomenti trattati nel corso

5. Modalità d'insegnamento:

Le lezioni sono di tipo frontale in aula con esempi dal PC del docente collegato in rete e al proiettore durante la lezione.

Inoltre per la prima parte si prevede l'uso della lavagna tradizionale per sviluppo di modelli analitici e calcoli.

6. Attività di supporto:

Seminari svolti da esperti del settore attivi nel mondo dell'ICT

Esame di articoli e survey presi dalla letteratura e indicati di volta in volta dal docente.

Uso della piattaforma Moodle.

7. Programma:

Prima parte:

- Principi fondamentali: architettura end-to-end, nomi e indirizzi,

segnalazione, segmentazione, randomizzazione, indirezione, multiplazione, virtualizzazione, scalabilità.

- Allocazione delle risorse di rete: ingegneria del traffico, controllo di congestione come un problema di allocazione delle risorse, TCP.
- Router design: Code input/output, la classificazione dei pacchetti,

scheduling

- Content Centric Networks: reti di distribuzione di contenuti, nomi vs indirizzi, reti di cache.
- BitTorrent: specifiche del protocollo, modelli prestazionali.
- Reti cellulari: architettura e principi, l'evoluzione degli standard.
- Reti di sensori: principi fondamentali e problematiche principali.
- Reti wireless: complementi su 802.11, Bluetooth, reti ad-hoc, routing e scheduling opportunistici.

Seconda parte: Parte istituzionale su Identity and Access Management Parte monografica su Buffer Overflow & Stack Abuse

8. Testi consigliati e bibliografia:

Prima parte:

Reti di calcolatori e internet. Un approccio top-down, 6/Ed. James F. Kurose, Keith W. Ross, Pearson Education Italia, 2013

Communication Networks An Optimization, Control and Stochastic Networks Perspective, R. Srikant, Lei Ying, Cambridge University Press, 2014

Seconda parte:

Libro di testo: William Stallings: Cryptography and Network Security Prentice Hall, Seconda Edizione, 1998

Altri Testi per consultazione: David Curry: Unix System Security, Addison-Wesley, 1992.

Insegnamento**MFN0946 - Economia e Gestione delle Imprese
Net Based**

Insegnamento (inglese):	Management of Net Based Enterprises
CFU:	6
Settore:	SECS-P/08 - ECONOMIA E GESTIONE DELLE IMPRESE
Periodo didattico:	2
Tipologia di Attività Formativa:	C - affine e integrativa
Docenti:	Marco PIRONTI (Titolare) Paola Maria PISANO (Titolare)

1. Prerequisiti e Propedeuticità:**Competenze attese in ingresso**

principi e tecniche del business planning e dell'economia aziendale

Eventuali corsi propedeutici

Economia e gestione delle imprese

2. Obiettivi formativi:

analizzare in maniera critica i nuovi modelli di business legati alle tecnologie della comunicazione e dell'informazione

3. Risultati dell'apprendimento attesi:

modelli e tecniche di analisi dei nuovi modelli di business

4. Modalità di verifica dell'apprendimento:

La prova d'esame Ã¨ scritta. L'esame verterÃ sui riferimenti bibliografici, sulle slides di supporto e sui cases study proposti. In riferimento ai cases study proposti, il candidato devra' produrre una relazione scritta (almeno 3 cartelle per caso proposto) ed inviarla via mail almeno una settimana prima dell'appello.

5. Modalità d'insegnamento:

lezioni frontali e working group su cases study

6. Attività di supporto:

Slides di supporto e colloqui individuali

7. Programma:

Aspetti generali della net-economy Le nuove fonti di vantaggio competitivo Modelli a supporto della strategia Internet business model. Analisi settoriale e casi pratici. Internet business e value generation. La nuova "value chain" SWOT Modello delle 5 forze La comunicazione d'impresa: le nuove opportunità . L'organizzazione aziendale: i nuovi paradigmi. Aspetti strategici: analisi interne ed esterne. Aspetti economici: analisi di fattibilitÃi, outsourcing. Aspetti finanziari: incubator, business angel, venture capitalist.

8. Testi consigliati e bibliografia:

Pironti M. "E-business models" Cedam 2002 Materiale fornito su e-learn

Insegnamento**MFN0972 - Elaborazione di Immagini e Visione Artificiale**

Insegnamento (inglese):	Image Processing and Artificial Vision
CFU:	9
Settore:	INF/01 - INFORMATICA
Periodo didattico:	1
Tipologia di Attività Formativa:	B - caratterizzante
Docenti:	Nello BALOSSINO (Titolare) Davide CAVAGNINO (Titolare) Marco GRANGETTO (Titolare)

1. Prerequisiti e Propedeuticità:**Competenze attese in ingresso**

Le lezioni presuppongono la conoscenza del calcolo vettoriale, matriciale e delle tecniche analitiche. La parte sperimentale richiede competenze di programmazione.

2. Obiettivi formativi:

I temi affrontati nel corso hanno lo scopo di fornire allo studente gli strumenti matematici ed informatici utili per l'elaborazione di immagini sia nel dominio spaziale sia in quello delle frequenze. Gli ambiti di elaborazione riguardano il miglioramento e ripristino di qualità, il riconoscimento di forme e i sistemi di visione artificiale. Il corso prevede anche una parte sperimentale basata sull'utilizzo degli ambienti MATLAB, EidosLab.

3. Risultati dell'apprendimento attesi:

Al termine del corso lo studente è in grado di utilizzare algoritmi atti al miglioramento di qualità di un'immagine nel dominio spaziale e in quello trasformato, alla descrizione di oggetti e al loro riconoscimento. Lo studente sarà inoltre in grado di analizzare e progettare sistemi di visione artificiale.

4. Modalità di verifica dell'apprendimento:

Svolgimento di un progetto per verificare la capacità dello studente a condurre insiemi di elaborazioni mirate a raggiungere scopi predefiniti e discussione sui temi trattati nel corso.

5. Modalità d'insegnamento:

Le lezioni in aula si svolgono sia in modalità tradizionale, sia attraverso l'ausilio di strumenti multimediali. Lo studente è invitato a interagire con il docente e alcune attività sperimentali sono volte ad approfondire la comprensione degli argomenti presentati a lezione.

6. Attività di supporto:

Vengono forniti dispense ed esempi di programmi in MATLAB visti durante il corso.

Tutto il materiale è scaricabile dalla piattaforma Moodle accessibile a partire dal sito I-learn (<http://www.educ.di.unito.it/>).

7. Programma:

Introduzione. Elaborazione di Immagini: definizioni. Esempi di ambiti d'uso dell'elaborazione di immagini. Passi fondamentali nell'elaborazione di immagini. Componenti di un sistema per l'elaborazione di immagini.

Immagini digitali: fondamenti. La luce e lo spettro elettromagnetico. Acquisizione e rappresentazione di immagini. Risoluzione spaziale e radiometrica. Una panoramica degli strumenti matematici usati nell'elaborazione di immagini.

Trasformazioni di luminosità e filtraggio spaziale. Esempi di trasformazioni e di filtraggio spaziale. Elaborazione dell'istogramma. Matching dell'istogramma. Elaborazioni locali dell'istogramma. Fondamenti di filtraggio spaziale. Filtri spaziali di smoothing. Filtri spaziali di evidenziazione dei contorni. Combinazione dei metodi di miglioramento basati sullo spazio.

Filtraggio nel dominio delle frequenze. Background. La trasformata di Fourier di funzioni di una variabile continua. Campionamento e la trasformata di Fourier di funzioni campionate. Estensione a funzioni di due variabili. La trasformata discreta di Fourier 2D e alcune sue proprietà. Fondamenti di filtraggio nel dominio delle frequenze. Smoothing di immagini usando filtri nel dominio delle frequenze. Evidenziazione dei contorni di immagini usando filtri nel dominio delle frequenze. La Fast Fourier Transform.

Miglioramento e ricostruzione di immagini. Riduzione del rumore periodico mediante filtraggio nel dominio delle frequenze. Degradazione lineare e non dipendente dalla posizione. Filtraggio inverso. Ricostruzione di immagini da proiezioni.

Elaborazione di immagini a colori. Fondamenti sui colori. Elaborazione di immagini a pseudo-colori. Trasformazioni di colori.

Elaborazione morfologica di immagini. Erosione e dilatazioni. Alcuni semplici algoritmi morfologici.

Segmentazione di immagini. Riconoscimento di punti, linee e confini di regioni. L'edge detector di Canny. Sogliatura. Segmentazione basata su aree. Divisione e unione di aree.

Rappresentazione e descrizione. Codici a catena. Segnature. Scheletri. Descrittori di confini. Descrittori di aree. Uso delle Componenti Principali per la descrizione.

Visione robotica. Elementi di percezione visiva. Visione stereoscopica.

8. Testi consigliati e bibliografia:

R. C. Gonzalez, R. E. Woods, "Digital Image Processing", Third ed., Pearson Education, 2008.

R. C. Gonzalez, R. E. Woods, S. L. Eddins, "Digital Image Processing Using MATLAB", Second ed., Gatesmark Publishing, 2009.

W. K. Pratt, "Digital Image Processing", Third ed., John Wiley and Sons, 2001.

J. C. Russ, "The Image Processing Handbook", CRC Press, 2007.

Insegnamento**MFN0974 - Elaborazione Digitale Audio e Musica**

Insegnamento (inglese):	Digital sound and music processing
CFU:	6
Settore:	INF/01 - INFORMATICA
Periodo didattico:	2
Tipologia di Attività Formativa:	B - caratterizzante
Docenti:	Vincenzo LOMBARDO (Titolare) Andrea VALLE (Titolare)

1. Prerequisiti e Propedeuticità:**Competenze attese in ingresso**

Nessuna

Eventuali corsi propedeutici

Nessuna

2. Obiettivi formativi:

I concetti fondamentali dell'acquisizione, rappresentazione, elaborazione, compressione del suono mediante strumenti informatici. Il protocollo MIDI per la rappresentazione della musica a livello simbolico. Tecniche di sintesi digitale del segnale audio e alla composizione algoritmica, in una prospettiva di utilizzo a vasto raggio, che comprenda l'ambito musicale ma anche le applicazioni al sound design e alla multimedialità. Linguaggi di programmazione e editor per l'elaborazione digitale del segnale audio.

3. Risultati dell'apprendimento attesi:

Conoscenze fondamentali di acquisizione, rappresentazione, elaborazione, compressione del suono mediante strumenti informatici. Conoscenza del protocollo MIDI. Tecniche di sintesi digitale del segnale audio. Linguaggi di programmazione e editor per l'elaborazione digitale del segnale audio.

4. Modalità di verifica dell'apprendimento:

Colloquio orale Prova di laboratorio

Per sostenere la prova orale, prenotarsi alla seguente pagina, selezionando "Facoltà di Scienze della Formazione", "corso di Studi DAMS", "attività didattica Tecnologie informatiche per il suono e l'immagine" e scegliere la data d'esame: <http://www.unito.it/unitoWAR/page/i...>

5. Modalità d'insegnamento:

Lezioni frontali di teoria complementate da esercizi assegnati sui software applicativi

6. Attività di supporto:

Esercizi di tecnica del suono sul web

7. Programma:

Il corso consta di due moduli. Il primo modulo è un'introduzione ai concetti fondamentali dell'acquisizione, rappresentazione, elaborazione, compressione del suono mediante strumenti informatici. Inoltre viene trattato il protocollo MIDI per la rappresentazione della musica a livello simbolico. Si articola in lezioni frontali e esercitazioni guidate in laboratorio. Il secondo modulo si

propone di fornire una introduzione alle tecniche di sintesi digitale del segnale audio e alla composizione algoritmica, in una prospettiva di utilizzo a vasto raggio, che comprenda l'ambito musicale ma anche le applicazioni al sound design e alla multimedialità. A tal proposito verrà introdotto il linguaggio di programmazione SuperCollider, che costituisce, allo stato attuale, lo strumento più potente e versatile per il controllo automatico della generazione di materiali sonori e della loro organizzazione.

<http://www.di.unito.it/~vincenzo/El...>

8. Testi consigliati e bibliografia:

V. Lombardo, A. Valle, Audio e multimedia, III edizione, Apogeo Editore, Milano, 2008 (capp. 1, 2, 3, 4, 5, 6, 8, 9).

A. Valle, SCIRMA: the SuperCollider Italian Manual at CIRMA, disponibile presso

<http://www.cirma.unito.it/andrea/sw...>

Insegnamento**MFN0970 - Fisica per Applicazioni di Realtà Virtuale**

Insegnamento (inglese):	Physics for Virtual Reality Applications
CFU:	6
Settore:	FIS/01 - FISICA SPERIMENTALE
Periodo didattico:	1
Tipologia di Attività Formativa:	C - affine e integrativa
Docenti:	Marco MAGGIORA (Titolare)

1. Prerequisiti e Propedeuticità:**Competenze attese in ingresso**

Basi di Matematica: equazioni, trigonometria, calcolo vettoriale, derivate ed integrali fondamentali.

Eventuali corsi propedeutici

nessuno

2. Obiettivi formativi:

Si intende fornire allo studente semplici applicazioni della Fisica nell'ambito delle conoscenze di base, dalla meccanica all'ottica, imparando a riconoscere i principi e le leggi fisiche sottostanti ed a sviluppare le equazioni necessarie per descriverle.

3. Risultati dell'apprendimento attesi:

Buona conoscenza di fenomeni fisici che appartengono alla vita di tutti i giorni o all'area delle grandi scoperte tecnologiche del XX secolo, per un utilizzo creativo nella produzione di tool didattici per la fisica.

4. Modalità di verifica dell'apprendimento:

Esame orale, con discussione delle applicazioni sviluppate durante il corso.

5. Modalità d'insegnamento:

Lezioni ed esercitazioni Durante le lezioni si utilizzano le slides (fornite in formato pdf) integrandole con spiegazioni alla lavagna; le slides, seppur esaustive, non esauriscono la totalità del programma, ma solo la sua grande maggioranza.

6. Attività di supporto:**7. Programma:**

1. Cinematica del punto materiale: moto rettilineo e moto circolare uniforme. 1.1 Spazio, velocità, accelerazione. 1.2 Elementi di calcolo vettoriale. 1.3 Moto rettilineo uniforme e uniformemente accelerato. 1.4 Moto parabolico (semplice e relativo). 1.5 Moto circolare, uniforme e non, orizzontale e verticale (semplice e relativo). 1.6 Moto relativo: moto della barca in un fiume; moto di un corpo da un veicolo in moto.

2. Fondamenti di dinamica. 2.1 Leggi della dinamica. 2.2 Gravitazione universale: legge di Newton, esperienza di Cavendish. 2.3 Velocità e distanza dalla terra per un satellite geostazionario 2.4

Sistemi inerziali e non inerziale: peso ed 'assenza di peso'. 2.5 Concetto di vincolo e di reazioni vincolari. 2.6 Piano inclinato.

3. Forze d'attrito. 3.1 Attrito statico e dinamico. 3.2 Equilibrio di un corpo su un piano inclinato scabro. 3.3 Moto circolare con attrito: forze agenti su un'automobile che percorre una curva, in piano o con curva sopraelevata; velocità limite.

4. Lavoro ed energia. Conservazione dell'energia meccanica. 4.1 Lavoro di forze costanti e variabili: definizione del segno del lavoro. 4.2 Energia cinetica. 4.3 Forze conservative, non conservative, e dissipative. 4.4 Energia potenziale (gravitazionale e elastica). 4.5 Conservazione dell'energia meccanica. 4.6 Moto di corpi utilizzando la conservazione dell'energia meccanica. 4.7 Effetti energetici della presenza di forze d'attrito.

5. Quantità di moto, impulso e conservazione della quantità di moto. 5.1 Quantità di moto. 5.2 Impulso e forze impulsive; teorema dell'impulso. 5.3 Quantità di moto per sistema a massa fissa ed a massa variabile: moto del razzo.

6. Urti elastici e anelastici. 6.1 Urti ed energia. 6.2 Pendolo balistico. 6.3 Urti in due dimensioni

7. Centro di massa. 7.1 Centro di massa: moto traslatorio; leggi della dinamica. 7.2 Centro di massa e baricentro. 7.3 Centro di massa del corpo umano.

8. Moto traslatorio e rotatorio. 8.1 Momento della forza e momento torcente. 8.2 Momento d'inerzia (calcolo per solidi semplici, tabelle per solidi particolari). 8.3 Leggi della dinamica rotazionale. 8.4 Moto di una carrucola. 8.5 Momento torcente e reazioni vincolari. 8.6 Energia cinetica rotazionale. 8.7 Moto traslatorio e rotatorio lungo un piano inclinato. 8.8 Moto roto-traslatorio e attrito. 8.9 Lavoro e potenza del momento torcente. 8.10 Conservazione del momento angolare. 8.11 Applicazione al corpo umano.

9. Statica e sistemi all'equilibrio. 9.1 Equilibrio di forze: applicazioni a muscoli e giunture. 9.2 Equilibrio di forze: applicazioni architettoniche.

10. Elasticità, sforzo e deformazione.

11. Moduli elastici, di taglio, di compressione, 11.1 Condizioni di equilibrio in trazione e compressione. 11.2 Applicazioni a muscoli e giunture. 11.3 Applicazioni architettoniche.

12. Moto armonico. 12.1 Oscillatore armonico semplice e forzato. 12.2 Considerazioni energetiche nel moto armonico.

13. Meccanica ondulatoria. 13.1 Natura e propagazione delle onde. 13.2 Onde meccaniche: trasversali, longitudinali e di superficie. 13.3 Energia, potenza e intensità delle onde. 13.4 Riflessione e trasmissione delle onde meccaniche.

14. Acustica. 14.1 Caratteristiche del suono. 14.2 Livello di intensità sonora. 14.3 L'orecchio umano. 14.4 Sensibilità acustica. 14.5 Interferenza. 14.6 Onde stazionarie. 14.7 Battimenti. 14.8 Acustica degli strumenti musicali: strumenti a corda e a fiato; analisi timbrica. 14.9 Effetti supersonici. 14.10 Effetto doppler. 14.11 Applicazioni ultrasoniche militari: sonar. 14.12 Applicazioni ultrasoniche mediche: ecografia, eco-doppler.

15. Ottica geometrica. 15.1 Modello a raggi. 15.2 Rifrazione e riflessione della luce; legge di Snell. 15.3 Formazione delle immagini: immagine reale e virtuale. 15.4 Specchi piani. 15.5 Specchi sferici: concavi e convessi; equazione degli specchi. 15.6 Effetti di rifrazione. 15.7 Effetti di riflessione totale: strumenti ottici prismatici e fibre ottiche. 15.8 Lenti sferiche sottili: divergenti e convergenti; equazione delle lenti. 15.9 Sistemi di lenti sottili. 15.10 Equazione del costruttore di lenti.

16. Ottica ondulatoria. 16.1 Principio di Huygens: definizione e legge di Snell. 16.2 Diffrazione e interferenza: fenditure di Young, diffrazione su oggetti e reticolo di diffrazione. 16.3 Fenomeni di dispersione: prisma e arcobaleno. 16.4 Ottica dei miraggi. 16.5 Interferenza su lamine sottili: bolle di sapone, anelli di Newton e rivestimento antiriflesso.

17. Applicazioni ottiche tecnologiche e medicali. 17.1 La macchina fotografica e gli obiettivi

fotografici. 17.2 L'occhio umano: cenni anatomici e analisi dei difetti visivi. 17.3 Applicazioni oculistiche. 17.4 Aberrazione di lenti e specchi: cenni. 17.5 Strumenti ottici complessi: lenti di ingrandimento, microscopio semplice e composto, cannocchiale, telescopio riflettore e rifrattore. 17.6 Limite di risoluzione di strumenti ottici.

8. Testi consigliati e bibliografia:

1. GIANCOLI Fisica Casa Editrice Ambrosiana oppure qualsiasi testo di Fisica I per Scienze o Ingegneria, come, ad esempio:

2. SERWAY Fisica 1 Edises 3. RESNICK, HALLIDAY, KRANE Fisica 1 Casa Editrice Ambrosiana

Insegnamento**INF0006 - Gestione delle Reti**

Insegnamento (inglese):	Networks Management
CFU:	6
Settore:	INF/01 - INFORMATICA
Periodo didattico:	2
Tipologia di Attività Formativa:	B - caratterizzante
Docenti:	Daniele MANINI (Titolare)

1. Prerequisiti e Propedeuticità:**Competenze attese in ingresso**

Lo Studente deve avere:

Una buona conoscenza delle tematiche e delle metodologie illustrate nell'indirizzo di Sistemi e Reti della Laurea Triennale in Informatica dell'Università di Torino;

Attenzione agli aspetti realizzativi, alla risoluzione dei problemi, all'integrazione sistemistica delle componenti, alla cooperazione e al lavoro di gruppo.

Eventuali corsi propedeutici

Corso di Laurea Triennale in Informatica, Indirizzo di Sistemi e Reti, Università di Torino.

2. Obiettivi formativi:

Gestione delle Reti (GR) e' un corso sui sistemi di gestione, che si concentra soprattutto sugli aspetti di gestione distribuita di sistemi distribuiti. L'aspetto preponderante e' quindi quello della comunicazione fra applicazioni e macchine, in primo luogo perche' un sistema distribuito deve essere gestito mediante una piattaforma di gestione che e' a sua volta distribuita sia sulle macchine sottoposte a gestione, sia sulle macchine usate come strumento di gestione; in secondo luogo perche' la comunicazione fra macchine ed applicazioni e' il componente essenziale da gestire. Una parte introduttiva sara' dedicata alla tematica dei protocolli applicativi di routing.

3. Risultati dell'apprendimento attesi:

Ci si attende che lo Studente sappia:

configurare un agente di gestione; analizzare, progettare e realizzare piattaforme di gestione utilizzando SNMP; compiere una valutazione dello stato di un sistema distribuito basandosi sugli elementi raccolti dal sistema di gestione

4. Modalità di verifica dell'apprendimento:

L'esame è esclusivamente orale. Nell'esame si verifica la comprensione dei principi fondanti della Gestione di Reti illustrati a lezione, e la comprensione dei meccanismi di funzionamento e delle motivazioni, che sono alla base della realizzazione del protocollo, illustrato a lezione, e dei sistemi di gestione, quali quello utilizzato nel laboratorio.

5. Modalità d'insegnamento:

Le lezioni frontali sono svolte con l'ausilio di trasparenze e personal computer. Le trasparenze sono a disposizione degli studenti sul sito Internet del Corso di Laurea (in attesa delle nuove, sono visualizzate quelle dell'anno passato!). Il laboratorio è svolto utilizzando il sistema di gestione HP Open View su piattaforma Unix e Windows.

6. Attività di supporto:

Oltre alle trasparenze usate a lezione, che sono a disposizione degli studenti sul sito Internet del Corso di Laurea, lo studente può utilizzare l'aula Babbage attrezzata per didattica di Alta Qualità che è dotata di workstation Unix e Windows sulle quali è installato il sistema di gestione HP Open View, sia in versione client che server. L'uso di tale laboratorio fornisce strumenti di comprensione di quanto illustrato a lezione, e di validazione del percorso di apprendimento effettuato dallo Studente.

7. Programma:

Cosa è SNMP: Cosa vuol dire "gestire"; architettura di un sistema di gestione Il protocollo SNMP: SNMP e UDP, SNMP Communities, SMI, Naming dei Managed Object RFC 1213 MIB-II Il protocollo SNMP: Object Identifier, La definizione degli OID, I tipi di dato della SMIv1, Un frammento della MIB-II, Gli oggetti colonnari, Ancora qualcosa di MIB-II Il protocollo SNMP: I concetti di base; Community; Servizio di autenticazione; Politica di accesso; Servizio proxy; Identificazione delle istanze; Oggetti colonnari; Riferimenti ambigui alle righe; Ordine lessicografico; Specifica del protocollo SNMPv1: Formati SNMPv1; i TAG dell'ASN.1; Trattamento di un messaggio da spedire; Trattamento di un messaggio ricevuto; Variable Bindings; Recuperare oggetti sconosciuti; Frequenza di polling; Le limitazioni di SNMP(v1) Abstract Syntax Notation 1 - ASN.1: Sintassi astratta (e concetti fondamentali) Le operazioni SNMP: L'operazione get; l'operazione get-next. Le operazioni SNMP: L'operazione get-bulk; L'operazione set; Codici di errore per get, get-next, get-bulk, set; Le operazioni SNMP: L'operazione Trap; L'operazione Notification;

8. Testi consigliati e bibliografia:

William Stallings, "SNMP, SNMPv2, SNMPv3, and RMON 1 and 2", 3rd Edition, Addison-Wesley Pub Co; ISBN: 0201485346 Comer, Internetworking con TCP/IP. Vol. 1 - Principi, protocolli e architetture, 5a ed., Pearson Education Italia, ISBN 9788871922805

Insegnamento**MFN0942 - Intelligenza Artificiale e Laboratorio**

Insegnamento (inglese):	Artificial Intelligence (with experimentations)
CFU:	9
Settore:	INF/01 - INFORMATICA
Periodo didattico:	2
Tipologia di Attività Formativa:	B - caratterizzante
Docenti:	Gian Luca POZZATO (Titolare) Pietro TORASSO (Titolare)

1. Prerequisiti e Propedeuticità:**Competenze attese in ingresso**

Per essere proficuamente seguito, il corso richiede conoscenze di base di Intelligenza Artificiale in particolare per quanto riguarda la nozione di agente intelligente, le problematiche di risoluzione automatica di problemi e i concetti basilari di rappresentazione della conoscenza. Per quanto riguarda le competenze di sviluppo/implementazione è richiesta la conoscenza di tecniche di programmazione e le nozioni di base di ingegneria del software.

Eventuali corsi propedeutici

Il corso di "Sistemi Intelligenti" della laurea triennale (o il corso da esso mutuato di "Istituzioni di sistemi intelligenti" per la laurea magistrale) fornisce le conoscenze attese in ingresso. Gli studenti che non abbiano esperienza nel settore troveranno utile documentarsi con attenzione sul testo S. Russell, P. Norvig. Intelligenza Artificiale, Un approccio moderno, terza edizione, Pearson/Prentice Hall concentrando l'attenzione sui capitoli 2, 3, 6, 7, 8 e 9.

2. Obiettivi formativi:

Il corso ha l'obiettivo di approfondire le conoscenze di Intelligenza Artificiale con particolare riguardo alle capacità di un agente intelligente di fare inferenze sulla base di una rappresentazione esplicita della conoscenza sul dominio. Alle competenze metodologiche si affiancano competenze progettuali perché il corso prevede la sperimentazione di metodi di ragionamento basati sul paradigma della programmazione logica e lo sviluppo di un agente intelligente in grado di esibire sia comportamenti reattivi che deliberativi (utilizzando ambienti basati su regole di produzione).

3. Risultati dell'apprendimento attesi:

Al termine del corso lo studente deve aver acquisito sia competenze metodologiche relative a meccanismi di ragionamento (anche in presenza di incertezza), che confidenza con i formalismi per la rappresentazione della conoscenza e con le strategie per la pianificazione automatica. Mentre queste conoscenze possono essere ascritte alle problematiche di Know What, il corso attraverso la parte di laboratorio sviluppa anche le competenze di Know How, in particolare la capacità di progettare e implementare un agente intelligente con capacità sia deliberative che di monitoraggio del proprio piano di azione. In particolare, lo studente sperimenterà lo sviluppo di metodi intelligenti utilizzando strumenti basati sulla programmazione logica, su linguaggi a regole e sul paradigma dell'Answer Set Programming.

Nell'ambito del corso e dell'esame verrà incoraggiata e valutata la capacità di integrare le conoscenze metodologiche con quelle progettuali al fine di fare scelte motivate su quali strategie/metodologie/strumenti adottare, valutando i benefici e le criticità di tali scelte. La richiesta di presentare in sede d'esame una relazione sulla parte progettuale svolta presuppone che lo studente abbia maturato la capacità di comunicare le principali scelte effettuate, argomentando in modo chiaro le motivazioni di tali scelte.

Lo studio di problematiche complesse (di cui non sempre è disponibile un trade-off) fornisce le basi per una esperienza di apprendimento in gran parte autonomo, che sarà utile nelle fasi successive della carriera di ricerca o professionale.

4. Modalità di verifica dell'apprendimento:

L'esame consiste in una interrogazione individuale sugli argomenti trattati nella parte metodologica del corso, seguita dalla presentazione e discussione dei progetti (sviluppati anche in gruppo) proposti durante lo svolgimento del corso. L'esame prevede anche la dimostrazione on-line dei sistemi sviluppati. Prima dell'esame orale è necessario inviare con congruo anticipo (almeno 5 giorni prima della data concordata) ai docenti una relazione scritta, contenente una descrizione dei sistemi/esercizi sviluppati nonché una analisi critica dei risultati sperimentali ottenuti.

5. Modalità d'insegnamento:

Le lezioni, che si svolgeranno in laboratorio, conterranno sia momenti di lezione tipicamente frontale, in cui il docente responsabile dell'argomento introdurrà gli argomenti relativi alla parte metodologica, sia momenti in cui il docente illustrerà strumenti, linguaggi ed ambienti software utili per lo sviluppo della parte sperimentale, sia momenti in cui gli studenti avranno modo di verificare di persona l'uso di tali strumenti e linguaggi nella soluzione dei problemi assegnati. Durante questa ultima fase viene incoraggiata la presentazione da parte degli studenti delle soluzioni (anche preliminari) a cui sono giunti e la loro discussione sotto la guida del docente.

6. Attività di supporto:

La documentazione di base relativa a CLIPS e a CLINGO e il materiale aggiuntivo del corso saranno resi disponibili dai docenti sulla piattaforma di I-learn man mano che gli argomenti verranno sviluppati a lezione.

7. Programma:

Come è evidente dal nome stesso del corso, il corso di Intelligenza Artificiale contiene sia una parte metodologica che una parte progettuale/sperimentale di laboratorio.

- Formalismi logici per la rappresentazione della conoscenza e ragionamento (3 CFU – docente Pozzato). Dal punto di vista metodologico saranno affrontate problematiche relative a: meccanismi di ragionamento per calcolo dei predicati del primo ordine, programmazione logica, ragionamento non monotono, answer set programming. Queste metodologie verranno affrontate dal punto di vista sperimentale con l'introduzione dei principali costrutti del Prolog, lo sviluppo di strategie di ricerca in Prolog e l'utilizzo dell'ambiente CLINGO nella risoluzione di problemi in cui sia necessaria l'applicazione di meccanismi di ragionamento non monotono e del paradigma dell'Answer Set Programming.
- Pianificazione automatica ed esecuzione nel mondo reale (2 CFU – Docente Torasso). Le problematiche affrontate a livello metodologico riguardano: rappresentazione delle azioni e degli obiettivi, metodi per la generazione automatica di piani (compresi piani condizionali e conformanti), strategie di ricerca on-line in ambienti parzialmente sconosciuti, monitoraggio dell'esecuzione del piano, ripianificazione.
- Ragionamento in presenza di incertezza (1CFU - Docente Torasso). La parte metodologica copre problematiche di ragionamento probabilistico, reti bayesiane e metodi di ragionamento su reti bayesiane.
- Sviluppo di un agente intelligente (3 CFU – Docente Torasso). Questa parte sperimentale prevede lo sviluppo di un agente in grado di esibire sia comportamenti deliberativi che reattivi in un ambiente parzialmente osservabile. La natura esatta del dominio applicativo sarà nota all'inizio del corso. Per lo sviluppo dell'agente viene introdotto l'ambiente CLIPS che fornisce un ambiente collaudato ed efficiente per sistemi a regole di produzione e viene mostrato come sia possibile sviluppare alcune funzionalità di base di risoluzione di problemi utilizzando il motore inferenziale di CLIPS.

8. Testi consigliati e bibliografia:

- S. J. Russell e P. Norvig, *Artificial Intelligence: a modern approach*, Third edition, 2010, Pearson (esiste anche traduzione italiana della prima parte della terza edizione e la traduzione integrale della seconda edizione, entrambe editate da Pearson/Prentice Hall).

Verranno affrontate le problematiche descritte nei capitoli 4, 9, 10,11, 13 e 14 delle terza edizione

- A. Dovier e A. Formisano. *Programmazione Dichiarativa in Prolog, CLP e ASP*. Disponibile gratuitamente all'indirizzo <https://users.dimi.uniud.it/~agostino.dovier/DID/Inc.pdf>.

Verranno affrontate le tematiche descritte nei capitoli 4, 5, 7, 8, 9, 10, 11, 12.

La documentazione di base relativa a CLIPS e a CLINGO verrà resa disponibile sulla piattaforma I-Learn.

Insegnamento**MFN1476 - Istituzioni di Basi di Dati**

Insegnamento (inglese):	Additional Databases
CFU:	6
Settore:	INF/01 - INFORMATICA
Periodo didattico:	2
Tipologia di Attività Formativa:	D - libera
Docenti:	Giuseppina Barbara DEMO (Titolare) Ruggero Gaetano PENSA (Titolare)

1. Prerequisiti e Propedeuticità:**Competenze attese in ingresso**

Lo studente deve avere familiarità con i concetti fondamentali della teoria degli insiemi e della logica. Deve aver acquisito capacità di progettare algoritmi fondamentali e programmare in linguaggi ad alto livello.

Eventuali corsi propedeutici

Insegnamenti di Logica, Matematica Discreta e Programmazione 1 e 2

2. Obiettivi formativi:

L'insegnamento è un'introduzione alle basi di dati e ai sistemi di gestione delle medesime (SGBD). Si propone perciò di fornire allo studente le prime competenze teoriche e pratiche sul modello relazionale e sull'architettura dei SGBD introducendo le componenti fondamentali. In aula saranno introdotti i concetti teorici di base. Gli studenti acquisiranno conoscenze (teoriche e pratiche) su: modello relazionale, algebra relazionale, dipendenze funzionali e normalizzazione, elementi di architettura dei SGBD relazionali (concorrenza, ripristino e buffer management, dizionario dei dati, memorizzazione efficiente dei dati).

3. Risultati dell'apprendimento attesi:

Gli studenti devono aver assimilato i concetti:

- i fondamenti matematici del modello relazionale,
- le interrogazioni in algebra ed in calcolo relazionale,
- i fondamenti della teoria della normalizzazione,
- le strutture dati per la memorizzazione efficiente dell'informazione.

4. Modalità di verifica dell'apprendimento:

E' obbligatoria l'iscrizione all'ambiente di e-learning Moodle dove gli studenti troveranno forum di discussione, materiale didattico, compiti assegnati e quiz sugli argomenti delle lezioni preparati dai docenti. La modalità d'esame comprende una prova scritta. La prova scritta prevede da 6 a 8 domande in tutto, al fine di verificare l'apprendimento dei concetti di base. Le domande riguardano argomenti ed esercizi della parte di teoria. La valutazione è in trentesimi.

5. Modalità d'insegnamento:

Lezioni frontali: 60 ore. La frequenza costante è caldamente consigliata.

6. Attività di supporto:

E' obbligatoria l'iscrizione all'ambiente di e-learning Moodle dove gli studenti troveranno forum di discussione, materiale didattico e compiti assegnati. E' altresì obbligatoria la consultazione di tali

materiali.

7. Programma:

Funzionalità e componenti dei sistemi di gestione di basi di dati. Fondamenti teorici delle basi di dati relazionali:

- il modello relazionale delle basi di dati (definizioni, proprietà principali, vincoli di integrità),
- algebra relazionale,
- introduzione al calcolo relazionale,
- dipendenze funzionali e teoria della normalizzazione,
- memorizzazione efficiente dei dati (B+ alberi),
- cenni alle tecniche di ottimizzazione,
- Introduzione alle transazioni: problemi di concorrenza e di affidabilità, livelli di isolamento.

Progettazione e programmazione delle basi dati:

- Introduzione alle metodologie di progettazione del software e loro relazione con la progettazione della basi di dati, argomenti non trattati e relazione con altri insegnamenti (es. Ingegneria del Software).
- Considerazioni sui parametri quantitativi dello schema logico.
- Considerazioni su meccanismi d'indicizzazione.

8. Testi consigliati e bibliografia:

Testo di riferimento: Atzeni, Ceri, Fraternali, Paraboschi, Torlone, "Basi di dati - Modelli e linguaggi di interrogazione", McGraw-Hill, Quarta edizione, 2013.

Testo consigliato: Atzeni, Ceri, Fraternali, Paraboschi, Torlone: Basi di dati - Architetture e linee di evoluzione 2/ed, McGraw-Hill, 2007

Materiali aggiuntivi sono forniti dai docenti.

Insegnamento**MFN1477 - Istituzioni di Basi di Dati**

Insegnamento (inglese):	Additional Databases
CFU:	9
Settore:	INF/01 - INFORMATICA
Periodo didattico:	2
Tipologia di Attività Formativa:	D - libera
Docenti:	Luca ANSELMA (Titolare) Sara CAPECCHI (Titolare) Giuseppina Barbara DEMO (Titolare) Ruggero Gaetano PENSA (Titolare)

1. Prerequisiti e Propedeuticità:**Competenze attese in ingresso**

Lo studente deve avere familiarità con i concetti fondamentali della teoria degli insiemi e della logica. Deve aver acquisito capacità di progettare algoritmi fondamentali e programmare in linguaggi ad alto livello.

Eventuali corsi propedeutici

Insegnamenti di Logica, Matematica Discreta e Programmazione 1 e 2

2. Obiettivi formativi:

L'insegnamento è un'introduzione alle basi di dati e ai sistemi di gestione delle medesime (SGBD). Si propone perciò di fornire allo studente le prime competenze teoriche e pratiche sul modello relazionale, sulla progettazione di una base di dati e sull'architettura dei SGBD introducendo le componenti fondamentali. In aula saranno introdotti i concetti teorici di base. Gli studenti acquisiranno conoscenze (teoriche e pratiche) su: progetto di una base dati, modello Entità Associazioni (EA) (o Entity Relationship), modello relazionale, algebra relazionale e SQL, dipendenze funzionali e normalizzazione, elementi di architettura dei SGBD relazionali (concorrenza, ripristino e buffer management, dizionario dei dati, memorizzazione efficiente dei dati). In laboratorio gli studenti lavoreranno al progetto di una base dati e ad esercitazioni su casi di studio in SQL.

3. Risultati dell'apprendimento attesi:

Gli studenti devono aver assimilato i concetti:

- i fondamenti matematici del modello relazionale,
- le interrogazioni in algebra ed in calcolo relazionale e in SQL,
- i fondamenti della teoria della normalizzazione,
- le strutture dati per la memorizzazione efficiente dell'informazione.

Inoltre lo studente deve:

- saper analizzare i requisiti per la progettazione di una base di dati per rappresentarli in forma di diagrammi Entity Relationship o Entità Associazioni (EA) essere in grado di trasformare schemi EA in schemi relazionali,
 - essere in grado di trasformare interrogazioni in forma testuale in interrogazioni SQL, essere in grado di eseguire semplici ristrutturazioni di schemi EA sulla base di dati quantitativi (ad esempio: numero stimato di record e criticità delle interrogazioni),
 - aver compreso i concetti e l'uso di transazioni.
-

4. Modalità di verifica dell'apprendimento:

E' obbligatoria l'iscrizione all'ambiente di e-learning Moodle dove gli studenti troveranno forum di discussione, materiale didattico, compiti assegnati e quiz sugli argomenti delle lezioni preparati dai docenti. La modalità d'esame comprende una prova scritta e una di laboratorio. Prova scritta: la prova prevede da 6 a 8 domande in tutto, al fine di verificare l'apprendimento dei concetti di base. Di queste, due domande riguardano argomenti di laboratorio (progettazione ed SQL), le restanti domande riguardano argomenti ed esercizi della parte di teoria. La valutazione è in trentesimi.

5. Modalità d'insegnamento:

Lezioni frontali: 60 ore. Attività di laboratorio: 30 ore. La frequenza costante è caldamente consigliata.

6. Attività di supporto:

E' obbligatoria l'iscrizione all'ambiente di e-learning Moodle dove gli studenti troveranno forum di discussione, materiale didattico e compiti assegnati. E' altresì obbligatoria la consultazione di tali materiali.

7. Programma:

Funzionalità e componenti dei sistemi di gestione di basi di dati. Fondamenti teorici delle basi di dati relazionali:

- il modello relazionale delle basi di dati (definizioni, proprietà principali, vincoli di integrità),
- algebra relazionale,
- introduzione al calcolo relazionale,
- introduzione a SQL (DDL e DML), Dizionario dei dati in un DBMS
- dipendenze funzionali e teoria della normalizzazione,
- memorizzazione efficiente dei dati (B+ alberi),
- cenni alle tecniche di ottimizzazione,
- Introduzione alle transazioni: problemi di concorrenza e di affidabilità, livelli di isolamento.

Progettazione e programmazione delle basi dati:

- Specifica d'interrogazioni e realizzazione in SQL (con esercitazioni in laboratorio su ORACLE).
- Dizionario dei dati (con esercitazioni in laboratorio su ORACLE).
- Definizione e ruolo di "database administrator".
- Cenni ai meccanismi d'autorizzazione offerti da SQL (possibilmente con esercitazioni in laboratorio su ORACLE).
- Introduzione alle metodologie di progettazione del software e loro relazione con la progettazione della basi di dati, argomenti non trattati e relazione con altri insegnamenti (es. Ingegneria del Software).
- Progettazione concettuale/logica, usando il modello ER (Entity Relationship) con eventuali esercizi di reverse modelling.
- Considerazioni sui parametri quantitativi dello schema logico.
- Considerazioni su meccanismi d'indicizzazione.

8. Testi consigliati e bibliografia:

Testo di riferimento: Atzeni, Ceri, Fraternali, Paraboschi, Torlone, "Basi di dati - Modelli e linguaggi di interrogazione", McGraw-Hill, Quarta edizione, 2013.

Testo consigliato: Atzeni, Ceri, Fraternali, Paraboschi, Torlone: Basi di dati - Architetture e linee di evoluzione 2/ed, McGraw-Hill, 2007

Materiali aggiuntivi sono forniti dai docenti.

Insegnamento**MFN1473 - Istituzioni di Calcolo Matriciale e Ricerca Operativa**

Insegnamento (inglese):	Additional Matrix Calculus and Operational Research
CFU:	6
Settore:	MAT/09 - RICERCA OPERATIVA
Periodo didattico:	1
Tipologia di Attività Formativa:	D - libera
Docenti:	Roberto ARINGHIERI (Titolare) Andrea Cesare GROSSO (Titolare)

1. Prerequisiti e Propedeuticità:**Competenze attese in ingresso**

Nessuna, a parte nozioni di base di algebra e insiemistica comuni nei programmi di matematica della scuola superiore.

Eventuali corsi propedeutici

Nessuno.

2. Obiettivi formativi:

Il corso si propone di fornire agli studenti nozioni generali di calcolo matriciale, algebra e geometria, e nozioni più specifiche di ricerca operativa.

Il calcolo matriciale è uno strumento fondamentale per il calcolo scientifico. La ricerca operativa studia modelli e metodi, basati sulle tecniche introdotte, per l'utilizzo ottimale di risorse scarse (in ambiti produttivi, finanziari, ecc.).

3. Risultati dell'apprendimento attesi:

Lo studente deve apprendere nozioni di geometria di base e le tecniche essenziali per manipolare vettori e matrici in spazi a n dimensioni.

Lo studente deve acquisire la capacità costruire modelli di programmazione lineare sia a variabili continue che a variabili intere partendo dall'enunciato di un problema reale, e la conoscenza dei meccanismi di base e la teoria relativa agli algoritmi che operano su tali modelli.

4. Modalità di verifica dell'apprendimento:

L'esame è costituito da una prova scritta di durata di almeno 2 ore seguita da una prova orale facoltativa.

Regole da seguire durante la partecipazione ad una prova scritta:

- E' vietata la comunicazione, sotto ogni forma, sia tra persone in aula che fuori dall'aula. La violazione di questa regola comporta l'annullamento della prova.
- E' permesso portare come materiale il formulario del corso.
- E' vietato portare ed utilizzare gli appunti.
- E' permesso portare la calcolatrice.
- Autovalutazione in 3 passi:
 1. Le soluzioni dell'esame saranno rese disponibili on-line subito dopo la prova scritta.
 2. Lo studente che ha consegnato l'esame ha 24 ore di tempo per decidere di ritirare la propria prova. In caso di problemi di connessione, e' possibile comunque inviare una email al docente di riferimento.

3. Gli studenti che non ritirano lo scritto consegnato, in caso di prova non superata (ovvero voto < 15), consumano 1 delle 3 consegne a disposizione sui 5 appelli disponibili.

Validità dei risultati ottenuti durante le prove che formano l'esame:

- Il voto ottenuto durante una prova rimane valido durante tutto l'Anno Accademico in cui la prova è stata sostenuta.
- La ripetizione di una prova, ovvero presenza effettiva all'appello anche in caso di ritiro, comporta l'annullamento dell'esito della prova precedente.

Regole per il calcolo del voto di un esame:

- La prova scritta viene valutata da 0 a 33 e si considera superata con voto uguali o superiori a 15
- La prova orale viene valutata da 0 a 4.
- Il voto dell'esame si ottiene come somma dei voti ottenuti nelle prove che lo compongono.

Note: Iscrizione e partecipazione:

- Prenotazione: prenotarsi in tempo tramite i servizi per studenti raggiungibili via web da www.educ.unito.it.
- Ritardatari: coloro che non sono riusciti ad iscriversi per tempo, possono segnalare la loro partecipazione all'esame scrivendo una e-mail al docente del proprio corso; lo studente risulta ammesso alla prova con riserva.
- Non partecipazione all'esame: in caso di prevista non partecipazione all'esame, si richiede di cancellare la prenotazione e/o di avvisare per email al docente; questo consente al docente stesso di accettare i ritardatari ammessi con riserva.

5. Modalità d'insegnamento:

Lucidi proiettati in aula e tradizionali spiegazioni alla lavagna.

6. Attività di supporto:

Verranno forniti appunti disponibili alla pagina I-learn del corso.

7. Programma:

1. Vettori e matrici. Operazioni fondamentali.
2. Cenni di geometria in spazi vettoriali.
3. Combinazioni lineari, indipendenza lineare. Sottospazi e basi.
4. Insiemi convessi, poliedri.
5. Programmazione lineare.
 - Modellazione.
 - Struttura della regione ammissibile. Soluzioni di base.
 - Algoritmo del simplesso.
 - Dualità.
6. Cenni di programmazione a variabili intere.

8. Testi consigliati e bibliografia:

R.J. Vanderbei, "Linear Programming", Kluwer Academic Publishers (solo per eventuali approfondimenti - reperibile in biblioteca)

Insegnamento**MFN1001 - Istituzioni di Economia e Gestione dell'Impresa**

Insegnamento (inglese):	Additional Management, business administration
CFU:	6
Settore:	SECS-P/08 - ECONOMIA E GESTIONE DELLE IMPRESE
Periodo didattico:	2
Tipologia di Attività Formativa:	C - affine e integrativa
Docenti:	Marco PIRONTI (Titolare) Paola Maria PISANO (Titolare)

1. Prerequisiti e Propedeuticità:**Competenze attese in ingresso**

Il corso non presuppone conoscenze iniziali specifiche

Eventuali corsi propedeutici

NESSUNO

2. Obiettivi formativi:

l'obiettivo del corso è analizzare le caratteristiche basi di una azienda: dal modello di business teorico alla creazione dei processi produttivi, alla comunicazione e vendita dei prodotti.

Verranno studiate le relazioni delle aziende all'interno della catena del valore (Clienti /fornitori) e i loro settori di riferimento. I framework teorici saranno poi applicati per l'analisi di aziende, settori e mercati innovativi al fine di valutare come vengono colte le opportunità che l'innovazione dà alle aziende e ipotizzare sviluppi futuri di modelli. durante il corso gli studenti si confronteranno con casi aziendali reali e con imprenditori. gli studenteti del corso potranno inoltre partecipare al progetto silicon valley study tour e passare una settimana in Silicon Valley dove conoscere le aziende più innovative dell'IT.

Gli studenti hanno inoltre l'opportunità di partecipare: www.ggi-academy.it/

3. Risultati dell'apprendimento attesi:

CREAZIONE E ANALISI DEI MODELLI DI BUSINESS CREAZIONE DELLE STRATEGIE AZIENDALI
ANALISI DEI COMPETITOR, DEL SETTORE, DEL MERCATO.

SOFT SKILL: LAVORO IN TEAM CAPACITA' DI PRESENTARE PROBLEM SOLVING

4. Modalità di verifica dell'apprendimento:

70% esame scritto sul materiale delle lezioni 30% lavoro di gruppo i non frequentanti: 70% sul materiale del sito+ libri 30% lavoro individuale.

5. Modalità d'insegnamento:

Frontali

6. Attività di supporto:**7. Programma:**

business model struttura dell'organizzazione struttura dei processi struttura delle funzioni
strategie analisi di settore analisi di mercato

8. Testi consigliati e bibliografia:

Creare modelli di business. Un manuale pratico ed efficace per ispirare chi deve creare o innovare un modello di business Autore Osterwalder Alexander; Pigneur Yves

Insegnamento**MFN0986 - Istituzioni di Interazione Uomo Macchina**

Insegnamento (inglese):	Additional Human-computer Interaction
CFU:	6
Settore:	INF/01 - INFORMATICA
Periodo didattico:	1
Tipologia di Attività Formativa:	D - libera
Docenti:	Giovanni SACCO (Titolare) Marino SEGNAN (Titolare)

1. Prerequisiti e Propedeuticità:**Competenze attese in ingresso**

Si richiede una buona conoscenza delle basi di dati (fornita dal corso di Basi di Dati), dell'analisi e della progettazione object-oriented (fornita dai corsi di Programmazione II e di Algoritmi e strutture dati) e dei fondamenti della programmazione distribuita (fornita dal corso di Programmazione III). Lo studente deve inoltre avere la capacità di scrivere, compilare e verificare la correttezza di programmi in Java.

Eventuali corsi propedeutici

Basi di dati e sperimentazioni, Algoritmi e sperimentazioni, Programmazione II and III.

2. Obiettivi formativi:

Nella progettazione e sviluppo di un'applicazione software si devono tenere in conto due aspetti fondamentali: (i) l'interazione con l'utente che, indipendentemente dall'efficacia delle funzionalità offerte da un'applicazione, influenza sensibilmente il suo successo in termini di utilizzo. L'applicazione deve essere usabile ed accessibile per permettere un uso agevole del software e prevenirne l'abbandono da parte degli utenti. (ii) l'implementazione dell'applicativo, che richiede di guardare "dietro all'interfaccia utente" per andare a fondo su aspetti architetturali e tecnologici che possono influenzare non solo le prestazioni dell'applicazione, la sua scalabilità e robustezza, ma anche le tipologie di servizio che possono effettivamente essere offerte.

Partendo da queste considerazioni, il corso si pone un duplice obiettivo: da una parte, fornire la conoscenza di base necessaria per il disegno corretto di interazioni uomo-macchina, che sta alla base della progettazione di applicazioni di ogni genere (web e non, mobili, etc.). Dall'altra, concentrandosi sulle applicazioni mobili, che sono oggetto specifico del corso, fornire la conoscenza di base necessaria per la progettazione e lo sviluppo di applicazioni interattive, accessibili da terminali mobili (come per esempio gli smart phone), e caratterizzate da una logica applicativa mediamente complessa.

Più precisamente, per quanto riguarda l'interazione uomo-macchina, verranno acquisite competenze sia teoriche sia pratiche nel disegno corretto di interazioni, con specifico riferimento alla progettazione user-centered. Per quanto riguarda gli aspetti architetturali e tecnologici, il corso tratterà dal punto di vista sia teorico che pratico la programmazione lato client per device mobili su piattaforma Android e lo sviluppo di interfacce mobili. Per permettere agli studenti di sperimentare le nozioni apprese durante le ore di teoria in aula il corso include una sostanziale parte di laboratorio. I temi introdotti durante il laboratorio corredano e integrano le conoscenze derivanti dalla parte teorica (knowledge and understanding) e permettono agli studenti di familiarizzare con le metodologie e tecnologie introdotte, anche investigando soluzioni alternative (applying knowledge and understanding).

Inoltre durante le ore di laboratorio è previsto lo sviluppo di un'applicazione realistica con

interfaccia mobile. La preparazione e la discussione del progetto di laboratorio sono inoltre volte a stimolare le capacità di organizzare il lavoro in piccoli gruppi (max 4 studenti), e poi di illustrare verbalmente le soluzioni adottate (communication skills).

3. Risultati dell'apprendimento attesi:

Lo studente dovrà avere assimilato: la progettazione user-centered, la prototipazione, la valutazione dell'usabilità e il disegno inclusivo per categorie speciali di utenti. Inoltre dovrà avere assimilato i concetti rilevanti per analizzare l'interazione: affordance, modelli concettuali, metafore, feedback, etc.

Lo studente acquisirà anche la conoscenza delle varie architetture di riferimento per lo sviluppo di applicazioni web e mobile, e dei loro pro e contro; inoltre lo studente acquisirà la conoscenza dei modelli più comunemente adottati per gestire dialoghi mediamente complessi e articolati tra utente e applicazione, e delle tecnologie attualmente utilizzate per l'implementazione delle applicazioni e delle loro interfacce (web e mobile). Lo studente dovrà essere in grado di sviluppare applicazioni lato client in ambiente Android, usando il linguaggio Python.

4. Modalità di verifica dell'apprendimento:

L'esame è composto da un TEST SCRITTO e da una VERIFICA DI LABORATORIO. Le due prove possono essere sostenute in qualsiasi ordine (cioè, non è necessario aver dato lo scritto per fare la prova di laboratorio, né il vice versa).

TEST SCRITTO: Prova scritta che include esercizi e domande teoriche sul programma del corso. Viene valutata da un minimo di 0 ad un massimo di 15 e si considerano sufficienti i voti ≥ 9 . Durante la prova è proibito comunicare con altre persone, presenti in aula o fuori. Inoltre, non si può portare alcun tipo di materiale didattico (appunti, libri, dispense, etc.) ed è vietato usare computer, telefonini o simili. Come da regolamento di Ateneo, ogni studente può sostenere un numero massimo di tre prove scritte durante l'Anno Accademico (cioè, consegnare il proprio elaborato tre volte). Il voto ottenuto durante un test scritto decade se lo studente partecipa ad un altro test scritto e consegna il suo elaborato.

VERIFICA DI LABORATORIO: prevede la discussione del progetto di laboratorio svolto durante il corso. La discussione deve essere effettuata preferibilmente in unica soluzione, con tutti i membri del gruppo di laboratorio presenti. Il voto di laboratorio è un numero compreso tra 0 e 15, si considerano sufficienti i voti ≥ 9 . CALCOLO DEL VOTO FINALE DI ESAME: Sia X il voto del test scritto; sia Y il voto di laboratorio. Il voto Fin finale dell'esame si ottiene come segue: $Fin = (X+Y)$

Note: i voti acquisiti durante la prova di laboratorio, o durante il test scritto, rimangono validi fino al termine della terza sessione d'esame (quella che precede l'inizio del nuovo corso). Quando si superano entrambe le prove, è necessario registrare il voto finale entro i limiti imposti dal Regolamento di Ateneo.

5. Modalità d'insegnamento:

L'insegnamento è diviso in una parte di teoria e una di laboratorio. Per la parte di teoria sono previste 30 ore di lezione frontali che seguono il programma riportato più avanti, integrate da casi di studio e da esercitazioni volte ad illustrare l'applicazione pratica dei concetti appena studiati.

La parte di laboratorio consiste di 30 ore ed è focalizzata sulla programmazione client-side in ambiente Android utilizzando il linguaggio Python e la libreria grafica Kivy. Le lezioni si svolgono in maniera interattiva e sono corredate da vari esercizi miranti a fornire esempi pratici.

Le sperimentazioni che vengono effettuate durante le ore di laboratorio, strutturate come sequenze di esercizi specifici, sono fondamentali per aiutare gli studenti a comprendere e assimilare i contenuti teorici spiegati a lezione in quanto permettono di mettere in pratica i concetti e le metodologie illustrate su esempi concreti. Inoltre lo sviluppo del progetto di laboratorio permette di consolidare le conoscenze teoriche in un caso realistico di media complessità.

Si consiglia caldamente la frequenza costante alle lezioni di teoria e di laboratorio. E' inoltre fondamentale che gli studenti si iscrivano al corso online su I-Learn, all'interno del quale i docenti

mettono a disposizione materiale didattico di supporto.

6. Attività di supporto:

Vengono forniti on-line i lucidi delle lezioni, link a documentazione disponibile sul Web ed alcuni testi degli esami scritti degli anni precedenti.

Il materiale didattico di supporto (lucidi delle lezioni, link a documentazione, esempi di testo di esami ed altro) è disponibile presso il supporto on-line ai corsi I-learn .

Si noti che i lucidi non sostituiscono il libro di testo, ne' il materiale integrativo ad esso affiancato.

7. Programma:

- Parte Ia - Human-computer interaction (HCI)
 - Human-computer interaction (HCI): Definizioni e contesto, evoluzione di HCI, nuove direzioni.
 - Il fattore umano: percezione (gestalt e affordance), attenzione e memoria, modelli mentali, metafore, il modello di Shneiderman e il modello di Norman.
 - Disegno di interazioni: user-centered design, requisiti funzionali e di usabilità (raccolta, analisi, presentazione), prototipazione, linee guida (con gestione degli errori ed assistenza agli utenti), elementi di tipografia elettronica, di layout e gestione del colore.
 - Tecniche di valutazione: valutazione senza utenti (quantitativa e qualitativa), valutazione con utenti, problemi, presentazione dei risultati.
 - Disegno inclusivo: accessibilità, disegno per utenti di differenti gruppi di età (bambini, anziani), internazionalizzazione.
- Parte Ib - Programmazione di device mobili.
 - Introduzione alla programmazione per mobile.
 - La piattaforma Android e sua architettura.
 - Processi e applicazioni in Android.
 - Il linguaggio Python e la libreria grafica Kivy. Le interfacce ai servizi di sistema.
 - Progettazione di una interfaccia utente in maniera programmatica e dichiarativa.
 - Sviluppare con Python o Java? Confronto tra gli ambienti ed esempi
 - Esempio di sviluppo del lato client di una semplice app per Android.

8. Testi consigliati e bibliografia:

Libri di Testo:

- Polillo, R. - FACILE DA USARE, Edizioni Apogeo, 2010
- Programmazione Web lato Server, di V. Della Mea, L. Di Gaspero, I. Scagnetto, Apogeo, 2007
- Android Training. <http://developer.android.com/training/index.html>
- Python. <http://pythonspot.com/beginner/>
- Kivy. <http://kivy.org/#home>

Altri testi (per consultazione):

- La caffettiera del masochista, di D. Norman, Apogeo

Insegnamento**MFN0985 - Istituzioni di Linguaggi Formali**

Insegnamento (inglese):	Additional Formal Languages
CFU:	6
Settore:	INF/01 - INFORMATICA
Periodo didattico:	1
Tipologia di Attività Formativa:	D - libera
Docenti:	Mario COPPO (Titolare) Maddalena ZACCHI (Titolare)

1. Prerequisiti e Propedeuticità:**Competenze attese in ingresso**

Lo studente deve avere familiarità con i concetti fondamentali della teoria degli insiemi e della progettazione di algoritmi iterativi e ricorsivi. Deve inoltre aver acquisito capacità di programmare in linguaggi ad alto livello.

Eventuali corsi propedeutici

Le competenze richieste per una proficua frequenza del corso sono fornite dagli insegnamenti: Programmazione I e laboratorio, Programmazione II e laboratorio, Architettura degli elaboratori, Matematica Discreta e Logica.

2. Obiettivi formativi:

Conoscenze nel campo della descrizione formale dei linguaggi e della traduzione (in particolare della compilazione) sono sempre state considerate fondamentali nel bagaglio culturale di un informatico e non possono essere ignorate dagli addetti al settore. Competenze di questo tipo si trovano nei curricula di orientamento informatico-matematico delle Università di tutto il mondo. I linguaggi di programmazione si sono evoluti presentando nuovi problemi di compilazione che hanno portato allo sviluppo di metodi generali per affrontarli. Buona parte della tecnologia di "front-end" dei compilatori, come grammatiche, espressioni regolari, parsificatori e traduttori guidati dalla sintassi, trovano anche applicazione in tutti i programmi in cui sia richiesta l'analisi strutturale di un testo o, in generale, di dati in cui si debba individuare una struttura.

L'insegnamento si propone pertanto di fornire allo studente una visione introduttiva dei problemi connessi alla definizione e alla traduzione dei linguaggi di programmazione, con particolare riferimento al progetto e alla costruzione di compilatori. Le metodologie e le tecniche presentate sono utili in generale come formalismi per definire il comportamento di un sistema o per realizzare traduttori più semplici di un compilatore vero e proprio.

3. Risultati dell'apprendimento attesi:

Comprensione e acquisizione della terminologia tecnica del settore. Conoscenza delle metodologie fondamentali per la descrizione della sintassi e della semantica dei linguaggi di programmazione e delle principali tecniche di parsificazione e traduzione; capacità di utilizzare tali conoscenze per lo sviluppo di sistemi software. Conoscenza e padronanza degli strumenti di base per la progettazione di traduttori.

4. Modalità di verifica dell'apprendimento:

Ai fini della determinazione del voto finale agli studenti viene proposto uno scritto, formato da: a) domande di teoria su tutto il programma svolto a lezione; b) esercizi che intendono verificare che lo studente sia in grado di applicare quanto appreso.

5. Modalità d'insegnamento:

Sono previste 60 ore di lezioni frontali. Nel corso delle lezioni è previsto lo svolgimento di esercizi in aula per ognuna delle parti fondamentali in cui si divide l'insegnamento: costruzione di automi, di grammatiche, definizione di parsificatori ascendenti o discendenti per semplici linguaggi, definizione di regole di traduzione e algoritmi di traduzione diretta dalla sintassi.

Le lezioni finali dell'insegnamento vengono dedicate alla preparazione dell'esame con svolgimento di esercizi tratti da testi d'esame di appelli degli anni precedenti.

Agli studenti sono resi disponibili i lucidi eventualmente usati dal docente come supporto alle lezioni.

6. Attività di supporto:

Il materiale didattico di supporto (programma dettagliato, lucidi, esempi di testi d'esame ed altro) è disponibile sul supporto on-line ai corsi I-learn.

7. Programma:

Automi a stati finiti ed espressioni regolari, analisi lessicale. Grammatiche e famiglie di linguaggi. Analisi sintattica: - Parsificazione top-down; - Parsificazione bottom-up. Traduzione diretta dalla sintassi. Generazione del codice intermedio.

Il programma dettagliato del corso sarà pubblicato sul supporto on-line ai corsi I-learn

8. Testi consigliati e bibliografia:

A. V. Aho, M. S. Lam, R. Sethi, J.D. Ullman, "Compilatori: Principi, tecniche e strumenti", Pearson Paravia Bruno Mondadori S.p.A., 2009, ISBN 978-88-7192-559-2.

John E. Hopcroft, R. Motwani, J. D. Ullman, "Automi, Linguaggi e Calcolabilità", Pearson-Addison Wesley, 2009, ISBN 978-88-7192-552-3.

Insegnamento**MFN0984 - Istituzioni di logica**

Insegnamento (inglese):	Additional Mathematical Logic
CFU:	6
Settore:	MAT/01 - LOGICA MATEMATICA
Periodo didattico:	1
Tipologia di Attività Formativa:	C - affine e integrativa
Docenti:	Felice CARDONE (Titolare) Luca MOTTO ROS (Titolare)

1. Prerequisiti e Propedeuticità:**Competenze attese in ingresso**

Conoscenza della terminologia di base relativa alle parti del discorso: nomi, verbi, proposizioni, aggettivi.

Eventuali corsi propedeutici

Nessuno.

2. Obiettivi formativi:

Il corso fornisce allo studente le basi del ragionamento formale, esercitandolo a controllare la verità di enunciati della logica del prim'ordine. Per ognuno di questi argomenti vengono messe in luce alcune applicazioni della matematica e della logica all'informatica.

3. Risultati dell'apprendimento attesi:

Lo studente dovrà dimostrare di sapere riconoscere ed utilizzare le principali tecniche di dimostrazione (diretta, per assurdo, per contrapposizione, per casi) in semplici dimostrazioni di proposizioni relative a strutture algebriche e relazionali. Dovrà essere in grado di utilizzare le principali forme del principio di induzione (ordinaria, forte, principio del minimo) in semplici dimostrazioni aritmetiche o relative alla sintassi formale dei linguaggi proposizionali e del prim'ordine. Dovrà dimostrare di essere in grado di formalizzare mediante formule logiche semplici asserzioni formulate in italiano. Dovrà inoltre dimostrare di conoscere la terminologia di base della teoria dei reticoli e delle algebre di Boole, con particolare attenzione ai concetti di atomo, insieme inferiore (o di ideale d'ordine), distributività assieme al loro utilizzo nella rappresentazione dei reticoli distributivi e delle algebre di Boole finite. Infine, lo studente dovrà dimostrare di riconoscere l'esistenza o meno di biezioni tra insiemi infiniti costruiti mediante le operazioni di unione disgiunta, prodotto cartesiano, insieme potenza, e insieme delle sequenze finite.

4. Modalità di verifica dell'apprendimento:

Le prove di esame sono scritte, a meno di motivata richiesta di esame orale da parte dei docenti. Le domande possono riguardare sia la teoria che lo svolgimento di esercizi. L'esito dell'esame è espresso in trentesimi.

5. Modalità d'insegnamento:

Le lezioni e le esercitazioni sono tutte condotte alla lavagna in aula.

6. Attività di supporto:

Disponibilità di un tutor.

7. Programma:

Tecniche di dimostrazione (12 ore) • Dimostrazione diretta, per assurdo, per contrapposizione; • Connettivi logici e loro significato in termini di condizioni di verità; • Tavole di verità e conseguenza logica tra proposizioni.

Formalizzazione (6 ore) • Linguaggi proposizionali e del prim'ordine: termini, quantificatori, alfabeto non logico, formule; • Schemi di traduzione da linguaggio naturale in linguaggi del prim'ordine (condizione sufficiente, necessaria, per tutti gli n abbastanza grandi, ci sono n arbitrariamente grandi,...)

Il principio di induzione (12 ore) • Forma ordinaria e forte del principio di induzione; • Principio del minimo; • Equivalenza tra forme del principio di induzione; • Induzione strutturale; • Ricorsione.

Reticoli (12 ore) • Ordinamenti parziali, con esempi; • Minimo confine superiore, massimo confine inferiore: reticoli; • Distributività; • Applicazioni: reticoli nell'architettura dell'informazione, analisi formale dei concetti.

Algebre di Boole (12 ore) • Algebra degli insiemi; • Algebre di Boole: definizioni; • Algebre di Boole come reticoli; • Cenni alla rappresentazione delle algebre di Boole finite: atomi; • Calcolo delle funzioni booleane e algebra della commutazione; • Algebra della logica: algebre di Boole e calcolo proposizionale.

Insiemi infiniti (6 ore) • Insiemi numerabili e più che numerabili: esempi; • Operazioni infinite: unioni e intersezioni, prodotti e somme, con le principali proprietà.

8. Testi consigliati e bibliografia:

F. Preparata, R.T. Yeh, "Introduzione alle strutture discrete", Boringhieri, 1976.
Andretta, Cardone: "Dispense di logica matematica", 2013.

Insegnamento**MFN0988 - Istituzioni di Programmazione Distribuita in Rete**

Insegnamento (inglese):	Additional Distributed Programming
CFU:	6
Settore:	INF/01 - INFORMATICA
Periodo didattico:	1
Tipologia di Attività Formativa:	D - libera
Docenti:	Liliana ARDISSONO (Titolare)

1. Prerequisiti e Propedeuticità:**Competenze attese in ingresso**

Si richiede una buona conoscenza dell'analisi e della progettazione object-oriented (fornita dai corsi di Programmazione II e di Algoritmi e strutture dati) e dei meccanismi di base e delle problematiche della programmazione concorrente (fornita dal corso di Sistemi Operativi). Lo studente deve inoltre avere la capacità di scrivere, compilare e verificare la correttezza di programmi in Java.

Eventuali corsi propedeutici

Programmazione II, Algoritmi e strutture dati, Sistemi Operativi.

2. Obiettivi formativi:

Lo sviluppo di software efficiente e scalabile presuppone la capacità di programmare applicazioni distribuite e concorrenti. In particolare, la programmazione distribuita in ambiente object oriented arricchisce le nozioni di base di programmazione concorrente sfruttando il paradigma ad oggetti per una più chiara scomposizione delle attività da eseguire in parallelo e loro attribuzione alle entità software di competenza, che possono essere modellate come oggetti distribuiti che offrono i relativi servizi. E' quindi fondamentale per il curriculum di un laureato in informatica acquisire competenze approfondite su tali temi.

Il corso si pone l'obiettivo di fornire la conoscenza di base necessaria per la programmazione di applicazioni object-oriented distribuite e concorrenti, usando linguaggi ad alto livello, attraverso (i) l'invocazione remota di metodi degli oggetti, e (ii) la programmazione di thread paralleli, cioè di processi "leggeri" che possono operare su uno o più processori all'interno della stessa applicazione principale. Altro obiettivo fondamentale del corso è la tecnica di programmazione ad eventi per la realizzazione di interfacce grafiche, che stanno alla base di tutte le applicazioni desktop e web basate su finestre. Tutte le conoscenze verranno fornite utilizzando il linguaggio Java come base per le spiegazioni e la sperimentazione.

Per permettere agli studenti di sperimentare le nozioni apprese durante le ore di teoria in aula il corso include una sostanziale parte di laboratorio. I temi introdotti durante il laboratorio corredano e integrano le conoscenze derivanti dalla parte teorica (knowledge and understanding) e permettono agli studenti di familiarizzare con le metodologie e tecnologie introdotte, anche investigando soluzioni alternative (applying knowledge and understanding). Inoltre durante le ore di laboratorio è previsto lo sviluppo di un'applicazione distribuita realistica con interfaccia grafica. La preparazione e la discussione del progetto di laboratorio sono inoltre volte a stimolare le capacità di organizzare il lavoro in piccoli gruppi (max 4 studenti), e poi di illustrare verbalmente le soluzioni adottate (communication skills).

3. Risultati dell'apprendimento attesi:

Lo studente acquisirà la conoscenza dei concetti che stanno alla base della programmazione di interfacce grafiche e di applicazioni distribuite e concorrenti in linguaggi ad alto livello. Lo

studente dovrà essere in grado di sviluppare applicazioni distribuite in ambiente Java, utilizzando RMI per l'invocazione remota di metodi, la programmazione a thread per la gestione del parallelismo e SWING per la realizzazione delle interfacce grafiche.

4. Modalità di verifica dell'apprendimento:

L'esame è composto da un TEST SCRITTO e da una VERIFICA DI LABORATORIO. Le due prove possono essere sostenute in qualsiasi ordine (cioè, non è necessario aver dato lo scritto per fare la prova di laboratorio, né il vice versa).

- TEST SCRITTO: Prova scritta che include esercizi e domande teoriche sul programma del corso. Viene valutata da un minimo di 0 ad un massimo di 30 e si considerano sufficienti i voti ≥ 18 . Durante la prova è proibito comunicare con altre persone, presenti in aula o fuori. Inoltre, non si può portare alcun tipo di materiale didattico (appunti, libri, dispense, etc.) ed è vietato usare computer, telefonini o simili. Come da regolamento di Ateneo, ogni studente può sostenere un numero massimo di tre prove scritte durante l'Anno Accademico (cioè, consegnare il proprio elaborato tre volte). Il voto ottenuto durante un test scritto decade se lo studente partecipa ad un altro test scritto e consegna il suo elaborato.
- VERIFICA DI LABORATORIO: prevede la discussione del progetto di laboratorio svolto durante il corso. La discussione deve essere effettuata preferibilmente in unica soluzione, con tutti i membri del gruppo di laboratorio presenti, e si può tenere sia durante gli appelli appositamente dedicati che su appuntamento (previa email al docente). Il voto di laboratorio è un numero compreso tra 0 e 30, si considerano sufficienti i voti ≥ 18 .
- CALCOLO DEL VOTO FINALE DI ESAME: sia X il voto del test scritto; sia Y il voto di laboratorio. Il voto Fin finale dell'esame si ottiene come segue: $Fin = (X+Y)/2$

Note: i voti acquisiti durante la prova di laboratorio, o durante il test scritto, rimangono validi fino al termine della terza sessione d'esame (quella che precede l'inizio del nuovo corso). Quando si superano entrambe le prove, è necessario registrare il voto finale entro i limiti imposti dal Regolamento di Ateneo.

5. Modalità d'insegnamento:

L'insegnamento è diviso in una parte di teoria e una di laboratorio. Per la parte di teoria sono previste 40 ore di lezione frontali che seguono il programma riportato più avanti, integrate da casi di studio e da esercitazioni volte ad illustrare l'applicazione pratica dei concetti appena studiati. La parte di laboratorio si tiene in altre 20 ore e ha per oggetto l'insegnamento delle tecnologie per lo sviluppo di applicazioni distribuite e concorrenti in Java (RMI, thread), con interfacce grafiche a finestre (SWING). Le lezioni si svolgono in maniera interattiva e sono corredate da vari esercizi miranti a fornire esempi pratici.

Le sperimentazioni che vengono effettuate durante le ore di laboratorio, strutturate come sequenze di esercizi specifici, sono fondamentali per aiutare gli studenti a comprendere e assimilare i contenuti teorici spiegati a lezione in quanto permettono di mettere in pratica i concetti e le metodologie illustrate su esempi concreti. Inoltre lo sviluppo del progetto di laboratorio permette di consolidare le conoscenze teoriche in un caso realistico di media complessità.

Si consiglia caldamente la frequenza costante alle lezioni di teoria e di laboratorio. E' inoltre fondamentale che gli studenti si iscrivano al corso online su I-Learn, all'interno del quale il docente mette a disposizione materiale didattico di supporto.

6. Attività di supporto:

Il materiale didattico di supporto (lucidi delle lezioni, link a documentazione, esempi di testo di esami ed altro) è disponibile presso il supporto on-line ai corsi: [qui \(PER ORA PAGINA DEL 2014/15\)](#)

Si noti che i lucidi non sostituiscono il libro di testo, né il materiale integrativo ad esso affiancato.

7. Programma:

Programmazione ad eventi in Java: programmare interfacce grafiche.

- Sorgenti di eventi, gestori di eventi, event-driven programming.
- Organizzazione e uso delle interfacce grafiche di Java.
- L'architettura Model-View-Controller (MVC).

Programmazione Multithread:

- Esecuzione concorrente di istruzioni.
- I thread in Java: ciclo di vita dei thread.
- Creazione e sincronizzazione di thread.
- Estensione del modello della memoria in presenza di thread.
- Problemi di sincronizzazione e loro risoluzione mediante il linguaggio Java.

Programmazione in rete in Java:

- L'architettura client-server.
- Uso di socket.
- Polimorfismo e trasferimento di oggetti mediante Java.
- Invocazione remota di metodi (RMI).
- Il modello di esecuzione distribuita di oggetti.

8. Testi consigliati e bibliografia:

Libro di Testo: Silvia Crafa. Oggetti, concorrenza, distribuzione. Società editrice Esculapio, 2014.

Altri testi (approfondimenti):

- K. Arnold, J. Gosling, D. Holmes. Il linguaggio Java, Manuale Ufficiale. Pearson Education Italia, 2006.
- Programmazione Java - tecniche avanzate, di Deitel e Deitel. Ed. Pearson - Prentice Hall

Insegnamento**MFN0987 - Istituzioni di Sistemi Intelligenti**

Insegnamento (inglese):	Additional Intelligent Systems
CFU:	6
Settore:	INF/01 - INFORMATICA
Periodo didattico:	2
Tipologia di Attività Formativa:	D - libera
Docenti:	Pietro TORASSO (Titolare)

1. Prerequisiti e Propedeuticità:**Competenze attese in ingresso**

Il corso di Istituzioni di Sistemi Intelligenti offerto nell'ambito delle Laurea Magistrale di Informatica è mutuato dal corso di Sistemi Intelligenti della laurea Triennale in Informatica.

Poichè Istituzioni di Sistemi Intelligenti è il primo corso che tratta argomenti di Intelligenza Artificiale, le competenze attese in ingresso riguardano competenze nel settore informatico. In particolare: - conoscenza di algoritmi su alberi e grafi con relative nozioni di complessità - esperienza di programmazione con particolare riferimento a programmazione ad oggetti (organizzazione in classi e sottoclassi, ereditarietà) - nozioni di logica (calcolo proposizionale e calcolo dei predicati del primo ordine) - nozioni di modelli semantici dei dati nelle basi dati.

Eventuali corsi propedeutici

Gli studenti che hanno conseguito la laurea in Informatica a Torino dovrebbero avere acquisito le competenze in ingresso sopra elencate seguendo gli insegnamenti di: - "Algoritmi e strutture dati", - "Programmazione I e II", - "Basi di dati", - "Matematica Discreta e Logica"

Corsi equivalenti sono (nella maggior parte dei casi) inseriti nell'offerta formativa di corsi di laurea in Informatica

2. Obiettivi formativi:

Il corso si propone di fornire una introduzione generale alle problematiche nel settore dell'Intelligenza Artificiale, con particolare attenzione a come sia possibile costruire un sistema dotato di capacità autonome di risoluzione di problemi, di ragionamento e di apprendimento quando abbia a disposizione una rappresentazione simbolica del mondo. Il corso si articola in tre parti principali: - Risoluzione automatica di problemi - Rappresentazione della conoscenza e ragionamento - Nozione di agente intelligente che agisce, ragiona ed apprende

Data la natura introduttiva del corso e la durata del corso, molte problematiche avanzate di Intelligenza Artificiale trovano collocazione nei corsi offerti per l'indirizzo "Sistemi per il Trattamento dell'Informazione" della laurea magistrale in Informatica.

L'insegnamento di Istituzioni di Sistemi Intelligenti è propedeutico per Intelligenza Artificiale e Laboratorio, e può essere di grande utilità per gli insegnamenti di Agenti Intelligenti, Sistemi Cognitivi e Apprendimento Automatico e analisi Intelligente dei Dati.

3. Risultati dell'apprendimento attesi:

Dal punto di vista della acquisizione di contenuti metodologici, gli studenti conosceranno le principali strategie di ricerca per la risoluzione automatica di problemi, i fondamenti della rappresentazione della conoscenza mediante formalismi logici e approcci strutturati (ontologie) e relativi meccanismi inferenziali.

Avranno inoltre acquisito le nozioni di base su architettura di un agente intelligente e su

apprendimento automatico di conoscenza a partire da esempi.

4. Modalità di verifica dell'apprendimento:

L'esame consiste in una interrogazione individuale sugli argomenti trattati nel corso. Tipicamente una delle domande richiede la capacità di applicare una metodologia/strategia per risolvere un semplice problema simile a quelli illustrati a lezione.

Pertanto nell'esame orale, oltre a verificare la conoscenza dei concetti e delle metodologie, si intende verificare la capacità di applicazione dei concetti e metodologie a problemi specifici.

Essendo l'esame un orale, è possibile inoltre verificare la chiarezza di esposizione e l'uso corretto della terminologia.

5. Modalità d'insegnamento:

Le lezioni si svolgono in aula. Oltre alle lezioni tipiche frontali in cui il docente illustra gli aspetti metodologici, nell'insegnamento numerose ore sono dedicate ad esempi ed esercitazioni (svolte dal docente). Vengono anche illustrati alcuni esempi significativi di strumenti software e di applicazioni di Intelligenza artificiale a problemi complessi del mondo reale. .

6. Attività di supporto:

Verrà reso disponibile sulla apposita pagina del corso all'interno del servizio I LEARN.

Tale materiale prevede ulteriori esempi di utilizzo dei meccanismi di ragionamento e di strategie di ricerca oltre a quelli contenuti sul libro di testo

7. Programma:

Come già detto l'insegnamento è una introduzione ai concetti basilari di Intelligenza Artificiale e si articola in tre parti strettamente connesse.

Parte 1) **RISOLUZIONE AUTOMATICA DI PROBLEMI** In questa parte si affronta la problematica di come definire il concetto di problema e di soluzione, di distinguere tra soluzione e soluzione ottima. Sono studiati tre approcci alla risoluzione di problemi: ricerca nello spazio degli stati, ricerca in spazi con avversario (giochi ad informazione completa), risoluzione di problemi mediante soddisfacimento di vincoli. Per ciascun approccio si discutono le principali strategie di ricerca: ampiezza, profondità, iterative deepening (per le ricerche cieche nello spazio degli stati), A* e Recursive Best First Strategy (per le ricerche euristiche), Min-Max e Alfa-beta (per i giochi con avversario), backtracking, forward propagation e arc consistency per meccanismi basati su soddisfacimento di vincoli. Particolare attenzione viene data alle garanzie offerte dalle diverse strategie in termini di qualità della soluzione e di complessità computazionale.

Parte 2) **RAPPRESENTAZIONE DELLA CONOSCENZA E RAGIONAMENTO** Il problema della rappresentazione della conoscenza e dei relativi meccanismi inferenziali viene affrontato studiando due principali famiglie di approcci alla rappresentazione della conoscenza: formalismi logici e rappresentazioni strutturate. Per quanto riguarda i formalismi logici si vede come sia il calcolo proposizionale che il calcolo dei predicati del primo ordine possano essere utilizzati per rappresentare conoscenza sul mondo e si vede come i meccanismi inferenziali (modus ponens, resolution, etc.) possano essere adoperati per fornire servizi utili (es. risposta a domande, verifica consistenza, ecc.). Si analizza anche come una rappresentazione a regole permetta meccanismi di ragionamento più efficienti (forward e backward chaining). Notevole attenzione viene data alla rappresentazione della conoscenza strutturata introducendo tassonomie, classi, individui, ereditarietà singola e multipla, inferenze specializzate. Queste nozioni vengono analizzate ed esemplificate mediante uso del linguaggio ontologico OWL2 (proposto e supportato da W3C).

Parte 3) **AGENTI E APPRENDIMENTO AUTOMATICO** In questa parte conclusiva si introduce la nozione di agente intelligente che opera in un ambiente e si fa vedere come l'agente possa avere sia comportamenti reattivi che deliberativi a seconda del compito assegnato. Si illustra come un agente debba avere capacità di risoluzione automatica di problemi e di ragionamento sullo stato del mondo e sul suo stato. Si descrive brevemente come l'apprendimento automatico sia una delle caratteristiche essenziali per ottenere un agente intelligente. Vengono introdotte solo nozioni

elementari con particolare riguardo all'apprendimento da esempi (in particolare apprendimento di alberi di decisione). Viene infine fatta una introduzione alle reti neurali come strumento per passare dal livello sub simbolico a quello simbolico.

8. Testi consigliati e bibliografia:

S. J. Russell e P. Norvig, *Artificial Intelligence: a modern approach*, Third edition, 2010, Pearson.

Nell'ambito del corso verranno affrontate le problematiche descritte nei capitoli 1,2,3,5,6,7,8,9, 18 del testo.

Si ricorda che esiste anche la traduzione italiana della prima parte della terza edizione (volume 1) e la traduzione integrale della seconda edizione (volume 1 e 2), entrambe edita da Pearson/Prentice Hall.

OWL 2 Web Ontology Language Primer (Second Edition) <http://www.w3.org/TR/owl2-primer/>

Insegnamento**MFN1474 - Istituzioni di Sistemi Operativi**

Insegnamento (inglese):	Additional Operating Systems
CFU:	6
Settore:	INF/01 - INFORMATICA
Periodo didattico:	1
Tipologia di Attività Formativa:	D - libera
Docenti:	Gianfranco BALBO (Titolare) Daniele GUNETTI (Titolare)

1. Prerequisiti e Propedeuticità:**Competenze attese in ingresso**

Agli studenti è richiesta una conoscenza di base dell'architettura di un computer (secondo quanto studiato nel corso di Architetture degli Elaboratori I) e dei concetti di base di programmazione (secondo quanto studiato nel corso di Programmazione I). Gli studenti dovranno inoltre essere in grado di padroneggiare i sistemi di enumerazione binario (base due) ed esadecimale (base sedici).

Eventuali corsi propedeutici

Costituiscono prerequisiti i contenuti dei corsi di:

* Architettura degli Elaboratori I * Programmazione I

2. Obiettivi formativi:

Il sistema operativo costituisce l'interfaccia fondamentale tra l'utilizzatore di un computer e il computer stesso. Parte essenziale del curriculum di base di un laureato in informatica è la conoscenza di come il sistema operativo sia in grado di amministrare le varie componenti hardware di cui è composto un computer. Queste modalità di amministrazione devono essere il più possibile trasparenti al generico utilizzatore del computer, ma devono essere conosciute a fondo da ogni specialista del settore. L'insegnamento fornisce dunque una conoscenza di base dell'architettura interna e del funzionamento dei moderni sistemi operativi, e di come, ai fini di garantire un ragionevole compromesso tra efficienza, sicurezza e facilità d'uso, vengono amministrate le risorse fondamentali della macchina su cui il sistema operativo è installato: il processore, la memoria principale e la memoria secondaria.

3. Risultati dell'apprendimento attesi:

Lo studente acquisirà la conoscenza dell'architettura e del funzionamento dei moderni sistemi operativi, e dovrà essere in grado di ragionare sulle prestazioni fornite dal sistema e su eventuali inefficienze. Avrà inoltre appreso i fondamenti della programmazione concorrente e la capacità di sviluppare programmi concorrenti in grado di interagire fra loro senza causare anomalie o blocchi del sistema. Infine, avrà una conoscenza di base di come le risorse della macchina (in particolar modo il tempo di CPU, lo spazio di memoria primaria e lo spazio di memoria secondaria) possano essere sfruttate al meglio.

4. Modalità di verifica dell'apprendimento:

Scritto con domande su tutti gli argomenti della parte di teoria del corso

5. Modalità d'insegnamento:

sono previste 60 ore di lezione frontali che seguono il programma riportato più avanti, integrate da casi di studio e da esercitazioni volte ad illustrare l'applicazione pratica dei concetti appena studiati.

6. Attività di supporto:

Dispense e lucidi usati a lezione, programmi di esempio.

- Per la parte di teoria del corso A il materiale è disponibile

[qui](#)

- Per la parte di teoria del corso B il materiale sarà disponibile in Moodle

7. Programma:

NOTA: Per la parte di teoria, il programma è basato sul TESTO DI RIFERIMENTO.

PARTE DI TEORIA:

* Introduzione al Corso di Sistemi Operativi

* PARTE I: GENERALITA'

o Introduzione (cap. 1)

o Strutture dei Sistemi Operativi (cap. 2)

* PARTE II: GESTIONE DEI PROCESSI

o Processi (cap. 3)

o Thread (cap. 4)

o Scheduling della CPU (cap. 5)

o Sincronizzazione dei Processi (cap. 6)

o Deadlock (Stallo di Processi) (cap. 7)

* PARTE III: GESTIONE DELLA MEMORIA (PRIMARIA)

o Memoria Centrale (cap. 8)

o Memoria Virtuale (cap. 9)

* PARTE IV: GESTIONE DELLA MEMORIA SECONDARIA

o Interfaccia del File System (cap. 10)

o Realizzazione del File System (cap. 11)

o Memoria Secondaria e Terziaria (Gestione dell'Hard disk)

8. Testi consigliati e bibliografia:

Teoria (TESTO DI RIFERIMENTO):

Silberschatz-Galvin-Gagne. Sistemi Operativi - ottava o nona edizione

Insegnamento**MFN1475 - Istituzioni di Sistemi Operativi**

Insegnamento (inglese):	Additional Operating Systems
CFU:	12
Settore:	INF/01 - INFORMATICA
Periodo didattico:	1
Tipologia di Attività Formativa:	D - libera
Docenti:	Marco ALDINUCCI (Titolare) Luca ANSELMA (Titolare) Gianfranco BALBO (Titolare) Luigi DI CARO (Titolare) Rossano GAETA (Titolare) Daniele GUNETTI (Titolare) Roberto MICALIZIO (Titolare) Daniele Paolo RADICIONI (Titolare)

1. Prerequisiti e Propedeuticità:**Competenze attese in ingresso**

Agli studenti è richiesta una conoscenza di base dell'architettura di un computer (secondo quanto studiato nel corso di Architetture degli Elaboratori I) e dei concetti di base di programmazione (secondo quanto studiato nel corso di Programmazione I). Gli studenti dovranno inoltre essere in grado di padroneggiare i sistemi di enumerazione binario (base due) ed esadecimale (base sedici).

Eventuali corsi propedeutici

Costituiscono prerequisiti i contenuti dei corsi di:

* Architettura degli Elaboratori I * Programmazione I

2. Obiettivi formativi:

Il sistema operativo costituisce l'interfaccia fondamentale tra l'utilizzatore di un computer e il computer stesso. Parte essenziale del curriculum di base di un laureato in informatica è la conoscenza di come il sistema operativo sia in grado di amministrare le varie componenti hardware di cui è composto un computer. Queste modalità di amministrazione devono essere il più possibile trasparenti al generico utilizzatore del computer, ma devono essere conosciute a fondo da ogni specialista del settore. L'insegnamento fornisce dunque una conoscenza di base dell'architettura interna e del funzionamento dei moderni sistemi operativi, e di come, ai fini di garantire un ragionevole compromesso tra efficienza, sicurezza e facilità d'uso, vengono amministrate le risorse fondamentali della macchina su cui il sistema operativo è installato: il processore, la memoria principale e la memoria secondaria.

Per la parte di laboratorio gli obiettivi formativi sono l'apprendimento del linguaggio C, utilizzato per la programmazione nell'ambiente del sistema operativo Unix. La parte di laboratorio mira a fornire allo studente una conoscenza (teorica e pratica) di base sui comandi della shell, sulla gestione dei processi, sugli strumenti di inter-process communication e sulla gestione dei segnali forniti dal sistema, oltre che alcuni rudimenti di programmazione bash.

I temi introdotti durante il laboratorio corredano e integrano le conoscenze derivanti dalla parte teorica (knowledge and understanding), al tempo stesso presentando esempi di problemi realistici di comunicazione e sincronizzazione su cui gli studenti sono sollecitati a cimentarsi, anche investigando soluzioni alternative (applying knowledge and understanding). La preparazione e la discussione del progetto sono inoltre volte a stimolare le capacità di organizzare il lavoro in piccoli gruppi (2-3 studenti), e poi di illustrare verbalmente le soluzioni adottate (communication skills).

- Corso A: TUTTE LE INFORMAZIONI SUL CORSO E IL MATERIALE DIDATTICO SI TROVERANNO ALL'URL:

[Sistemi Operativi](#)

- Corso B: TUTTE LE INFORMAZIONI SUL CORSO E IL MATERIALE DIDATTICO SI TROVERANNO IN MOODLE

3. Risultati dell'apprendimento attesi:

Lo studente acquisirà la conoscenza dell'architettura e del funzionamento dei moderni sistemi operativi, e dovrà essere in grado di ragionare sulle prestazioni fornite dal sistema e su eventuali inefficienze. Avrà inoltre appreso i fondamenti della programmazione concorrente e la capacità di sviluppare programmi concorrenti in grado di interagire fra loro senza causare anomalie o blocchi del sistema. Infine, avrà una conoscenza di base di come le risorse della macchina (in particolar modo il tempo di CPU, lo spazio di memoria primaria e lo spazio di memoria secondaria) possano essere sfruttate al meglio.

Lo studente dovrà essere in grado di sviluppare programmi scritti nel linguaggio C, e programmi di sistema e concorrenti codificati con i comandi propri dell'ambiente del sistema operativo Unix; dovrà inoltre essere in grado di ragionare su alcuni problemi di comunicazione fra processi e sincronizzazione utilizzando gli strumenti introdotti durante il laboratorio.

4. Modalità di verifica dell'apprendimento:

Ai fini della determinazione del voto finale l'esame dell'insegnamento di Sistemi Operativi è diviso in 3 parti, 2 obbligatorie ed una facoltativa.

1. Scritto obbligatorio. Permette di prendere un voto massimo di 27/30. Lo scritto è diviso a sua volta in due parti: a) domande di teoria su tutto il programma dell'insegnamento della parte di teoria, e in particolare sulla gestione dei processi, della memoria primaria e della memoria secondaria. Lo studente dovrà dimostrare di essere in grado di ragionare su situazioni specifiche, su casi di studio e su esempi numerici che possono rispecchiare problematiche reali dell'implementazione dei sistemi operativi e del loro funzionamento. b) domande sul laboratorio Unix e sul linguaggio C. 2. Discussione sull'esercitazione finale di laboratorio obbligatoria. Permette di incrementare di al massimo 3 punti il voto conseguito allo scritto (questa parte viene sostenuta oralmente con il docente del corso di laboratorio Unix, e deve essere sostenuta prima dell'esame scritto). 3. Orale facoltativo. Integra il voto ottenuto sostenendo le parti di esame 1) e 2), e permette di prendere un voto massimo di 30/30 e lode.

Per la parte di teoria, nel corso delle lezioni è previsto lo svolgimento di esercizi in aula per ognuna delle parti fondamentali in cui si divide l'insegnamento (gestione dei processi, gestione della memoria primaria, gestione della memoria secondaria). Alla fine dell'insegnamento vengono tenute alcune lezioni di preparazione all'esame in cui vengono svolti i test d'esame di appelli degli anni passati.

Per la parte di laboratorio, durante ciascuna lezione sono assegnati specifici esercizi; al termine di ogni macro-argomento sono proposte semplici esercitazioni riepilogative da svolgere in preparazione all'esercitazione finale, in modo da favorire l'assimilazione graduale degli argomenti introdotti durante il corso. L'esercitazione finale generalmente consiste in un progetto di maggiore impegno, e richiede di sviluppare un'applicazione in cui lo studente deve mostrare di padroneggiare i concetti e gli strumenti presentati durante il laboratorio, producendo inoltre una breve relazione in cui sono illustrati i problemi e le relative soluzioni.

La prova scritta comprende alcune domande sia sul linguaggio C sia sulla parte UNIX.

5. Modalità d'insegnamento:

L'insegnamento è diviso in una parte di teoria e una di laboratorio.

Per la parte di teoria sono previste 60 ore di lezione frontali che seguono il programma riportato più avanti, integrate da casi di studio e da esercitazioni volte ad illustrare l'applicazione pratica dei concetti appena studiati.

La parte di laboratorio è articolata in due moduli da 30 ore ciascuno, il primo avente per oggetto l'insegnamento del linguaggio C, e il secondo rivolto alla programmazione in ambiente UNIX. Le lezioni si svolgono in maniera interattiva e sono corredate da vari esercizi miranti a fornire esempi pratici.

6. Attività di supporto:

Dispense e lucidi usati a lezione, programmi di esempio.

- Per la parte di teoria del corso A il materiale è disponibile

[qui](#)

- Per la parte di teoria del corso B il materiale sarà disponibile in Moodle

7. Programma:

NOTA: Per la parte di teoria, il programma è basato sul TESTO DI RIFERIMENTO.

PARTE DI TEORIA:

* Introduzione al Corso di Sistemi Operativi

* PARTE I: GENERALITA'

o Introduzione (cap. 1)

o Strutture dei Sistemi Operativi (cap. 2)

* PARTE II: GESTIONE DEI PROCESSI

o Processi (cap. 3)

o Thread (cap. 4)

o Scheduling della CPU (cap. 5)

o Sincronizzazione dei Processi (cap. 6)

o Deadlock (Stallo di Processi) (cap. 7)

* PARTE III: GESTIONE DELLA MEMORIA (PRIMARIA)

o Memoria Centrale (cap. 8)

o Memoria Virtuale (cap. 9)

* PARTE IV: GESTIONE DELLA MEMORIA SECONDARIA

o Interfaccia del File System (cap. 10)

o Realizzazione del File System (cap. 11)

o Memoria Secondaria e Terziaria (Gestione dell'Hard disk)

=====

PARTE DI LABORATORIO:

- Linguaggio C
- Introduzione a Unix (comandi, shell, file system, diritti d'accesso, ridirezione, pipe)

- Make e makefile
- System call per la creazione e la sincronizzazione di processi
- System call per L'InterProcess Communication e per la gestione di segnali
- Esercitazioni pratiche, in particolare: esercitazioni finalizzate ad imparare il linguaggio C, ad utilizzare Unix e a sviluppare programmi concorrenti

8. Testi consigliati e bibliografia:

Teoria (TESTO DI RIFERIMENTO):

Silberschatz-Galvin-Gagne. Sistemi Operativi - ottava o nona edizione - Agli studenti sono anche resi disponibili fin dall'inizio del corso i lucidi usati dal docente nelle lezioni di teoria.

Laboratorio: • A. Kelley e I. Pohl: "C didattica e programmazione" (presente in Biblioteca) • R. Stevens, S. Rago: "Advanced Programming in the UNIX Environment", 2a ed., Addison Wesley • M.J. Bach: "The Design of the UNIX Operating System", Prentice-Hall International Editions (L3167 escluso dal prestito), per la parte relativa all'implementazione del kernel Unix. • S.R. Bourne: "The Unix System" (L1856 escluso dal prestito, L1803, L2140) • K. Christian: "The Unix Operating System", John Wiley & Sons (L3147 escluso dal prestito)

Insegnamento**MFN0989 - Istituzioni di Sviluppo Software**

Insegnamento (inglese):	Additional Software Engineering
CFU:	6
Settore:	INF/01 - INFORMATICA
Periodo didattico:	2
Tipologia di Attività Formativa:	D - libera
Docenti:	Viviana BONO (Titolare)

1. Prerequisiti e Propedeuticità:**Competenze attese in ingresso**

Conoscenza dei concetti studiati negli insegnamenti di Programmazione I e II.

Eventuali corsi propedeutici

Programmazione I e II.

2. Obiettivi formativi:

Il corso si propone di introdurre gli studenti allo sviluppo di applicazioni software, utilizzando la metodologia Agile Unified Process (UP), che sfrutta il linguaggio di modellazione UML. Lo studente dovrà saper sviluppare un'applicazione significativa individuando con chiarezza la logica applicativa, l'interazione con le basi di dati e le interfacce richieste dai requisiti. Inoltre dovrà imparare a pianificare il lavoro secondo i canoni dello sviluppo dei progetti: lavoro di gruppo, definizione degli obiettivi e delle fasi di sviluppo.

Il corso ha una forte caratterizzazione sperimentale. Per questa ragione è suddiviso in un modulo di 2 CFU di lezioni frontali e un modulo di laboratorio di 4 CFU.

3. Risultati dell'apprendimento attesi:

Lo studente acquisirà:

- conoscenza teorica e pratica della metodologia UP e dell'uso di UML in tale ambito;
- conoscenza di un tool a supporto della notazione UML;
- conoscenza della progettazione attraverso pattern di progettazione e pattern architetturali;
- capacità di presentare i risultati mediante una adeguata relazione.

4. Modalità di verifica dell'apprendimento:

L'esame consiste di due parti. La prima parte è rappresentata da un progetto di gruppo sviluppato in laboratorio, i cui risultati sono illustrati da una breve relazione. La valutazione del progetto è fatta mediante una discussione di gruppo, seguita da una discussione individuale. La valutazione di tale parte è espressa in trentesimi. La seconda parte consiste di una prova orale: su UML, UP e Design Pattern. La valutazione viene espressa in trentesimi. La votazione finale è data da una media pesata: la parte di laboratorio ha un peso di 2/3, la prova orale ha un peso di 1/3, la lode viene aggiunta se la media pesata è strettamente maggiore di 30. Le due parti si possono sostenere insieme (nello stesso appello) oppure separatamente, in qualsiasi ordine.

5. Modalità d'insegnamento:

NOTA: Esame mutuato da Sviluppo delle Applicazioni Software: la descrizione comprende anche la parte di "Elementi di ingegneria del software", che non viene richiesta per il presente esame (questa parte è infatti assente nelle sezioni precedenti del documento).

L'insegnamento è diviso in due parti di teoria e una di laboratorio.

Per la parte di teoria sono previste 40 ore di lezione frontali che seguono il programma riportato più avanti, integrate da casi di studio e da esercitazioni volte ad illustrare l'applicazione pratica dei concetti studiati.

La parte di laboratorio è composta da 50 ore e consiste nello sviluppo guidato di un progetto di medie dimensioni, realizzato in gruppi utilizzando la metodologia UP.

6. Attività di supporto:

Vengono effettuate consegne periodiche degli artefatti UP per il progetto di laboratorio, che man mano vengono corretti e commentati durante le ore di laboratorio.

7. Programma:

NOTA: Esame mutuato da Sviluppo delle Applicazioni Software: la descrizione comprende anche la prima parte di teoria, "Elementi di ingegneria del software", che non viene richiesta per il presente esame.

PRIMA PARTE DI TEORIA Elementi di ingegneria del software: modelli Waterfall, Spirale, V-shaped, Component-based Development, metodologie Agili tra cui: SCRUM e extreme programming, testing: unit testing, acceptance test, white e black box testing, controllo delle versioni.

SECONDA PARTE DI TEORIA Introduzione all'UML: use case diagram, class diagram, object diagram, sequence diagram, communication diagram, state chart, activity diagram. Una metodologia Agile: Unified Process (UP). Tale metodologia verrà descritta in dettaglio a lezione durante le ore di teoria e applicata a uno studio di caso nel laboratorio. I passi previsti, in breve, sono: Pianificazione delle fasi di sviluppo: esse sono suddivise in ideazione, elaborazione e costruzione, possiamo pensare ad esempio a una iterazione di ideazione, due iterazioni di elaborazione e una iterazione di costruzione. Di tutte le iterazioni si prevede una durata e i documenti prodotti (tabella delle attività). Prima iterazione (di ideazione). A partire da una descrizione informale del progetto da sviluppare, si comincia l'analisi dei requisiti con: scelta degli attori e descrizione dei loro obiettivi, individuazione dei casi d'uso (documento prodotto: use case diagram). Si fa poi una suddivisione fra i casi d'uso a seconda della loro priorità (ad esempio: alta, media, bassa), secondo dei criteri da fissare a priori. Si produce il class diagram di dominio. Si dettagliano i casi d'uso ad alta priorità, si fanno i sequence diagram di sistema per individuare le operazioni che corrisponderanno a eventi legati all'interfaccia utente e si scrivono i contratti di quelle operazioni che si ritengono più complesse. Si producono anche il documento di visione, il documento delle specifiche supplementari, la prima versione del glossario. Seconda e terza iterazione (di elaborazione). Durante queste iterazioni si eseguono sia attività di design e implementazione che di analisi. Per quanto riguarda la progettazione, si sceglie la classe Controller (la classe nel sistema che "parla" con l'interfaccia utente). Si dettagliano i corpi delle operazioni dei casi d'uso ad alta priorità tramite sequence o communication diagram. Si produce un secondo class diagram, in cui si dettagliano campi e metodi e si applicano i design pattern per organizzare il software. Si cominciano a progettare la base di dati e l'interfaccia utente. Si cominciano a sviluppare gli unit test e l'implementazione dei casi d'uso ad alta priorità. Per quanto riguarda l'analisi, si rivedono i casi d'uso sviluppati nel passo precedente. Si dettagliano i casi d'uso a priorità media e il class diagram di dominio. Si fanno i sequence diagram di sistema e i contratti per tali casi d'uso. Si aggiorna il glossario. Quarta iterazione (di costruzione). Si progettano e si implementano i casi d'uso a media priorità. Si rivedono eventualmente quelli ad alta priorità. Si procede con gli unit test (insieme eventualmente ad altri test). Si fa il deployment.

PARTE DI LABORATORIO Si veda la seconda parte di teoria.

8. Testi consigliati e bibliografia:

Larman. Applicare UML e i pattern. Terza edizione (seconda edizione italiana), Pearson (Capitoli 1-21; Capitoli 23-26; Cenni ai Capitoli 28, 29, 30, 31, 33, 34; Capitolo 36; cenni al Capitolo 38).

Altri testi consigliati:

Fowler. UML Distilled. Terza edizione italiana, Pearson-Addison Wesley.

Gamma, Helm, Johnson, Vlissides. Design pattern. Prima edizione italiana, Pearson Education Italia-Addison Wesley.

NOTE Software a disposizione nei laboratori: Modelio e Visual Paradigm.

Il materiale del corso consiste in slide, testi e soluzioni di esercizi, esempi, riferimenti bibliografici e articoli di approfondimento per la parte di teoria; in piu', per la parte di laboratorio, sono forniti artefatti completi e semi-lavorati appartenenti al progetto proposto. Tutto il materiale viene pubblicato sulla piattaforma I-learn (<http://informatica.i-learn.unito.it/>).

Insegnamento**MFN0990 - Istituzioni di Tecnologie Web**

Insegnamento (inglese):	Additional Web Technologies
CFU:	6
Settore:	INF/01 - INFORMATICA
Periodo didattico:	1
Tipologia di Attività Formativa:	D - libera
Docenti:	Liliana ARDISSONO (Titolare)

1. Prerequisiti e Propedeuticità:**Competenze attese in ingresso**

Si richiede una buona conoscenza delle basi di dati (fornita dal corso di Basi di Dati), dell'analisi e della progettazione object-oriented (fornita dai corsi di Programmazione II e di Algoritmi e strutture dati) e dei fondamenti della programmazione distribuita (fornita dal corso di Programmazione III). Lo studente deve inoltre avere la capacità di scrivere, compilare e verificare la correttezza di programmi in Java.

Eventuali corsi propedeutici

Basi di Dati, Interazione Uomo-Macchina, Programmazione II, Algoritmi e Strutture Dati, Programmazione III.

2. Obiettivi formativi:

Con la enorme diffusione delle applicazioni web diventa fondamentale nel curriculum di un laureato in informatica il conoscere a fondo le architetture delle stesse e le tecnologie più diffuse per il loro sviluppo. In particolare, l'implementazione dell'applicativo, che richiede di guardare "dietro all'interfaccia utente" per andare a fondo su aspetti architetture e tecnologici che possono influenzare non solo le prestazioni dell'applicazione, la sua scalabilità e robustezza, ma anche le tipologie di servizio che possono effettivamente essere offerte.

Il corso si pone come obiettivo di fornire la conoscenza di base necessaria per la progettazione e lo sviluppo di applicazioni Web interattive, accessibili da terminali desktop e mobili (grazie all'uso di linguaggi di interfaccia utente cross-platform), e caratterizzate da una logica applicativa mediamente complessa. In ultimo ci si propone di formare programmatori capaci di sviluppare applicazioni web di qualità e basate su architetture standard, largamente utilizzate nel mondo aziendale.

Le tecnologie presentate sono note come Server-side Programming e riguardano la progettazione e lo sviluppo di applicazioni basate su architetture modulari che possono accedere a sorgenti dati eterogenee (come basi dati relazionali, file, etc.) allo scopo di fornire all'utente servizi complessi. Più precisamente, il corso tratterà dal punto di vista sia teorico che pratico: (i) lo sviluppo di pagine web statiche e dinamiche con HTML5 e JavaScript, e (ii) la programmazione lato server, utilizzando tecnologie Web in ambiente Java. Inoltre, il corso tratterà la tecnologia XML, data la sua importanza nella condivisione di informazioni in Internet e nella gestione di interoperabilità tra applicazioni.

Per permettere agli studenti di sperimentare le nozioni apprese durante le ore di teoria in aula il corso include una sostanziale parte di laboratorio. I temi introdotti durante il laboratorio corredano e integrano le conoscenze derivanti dalla parte teorica (knowledge and understanding) e permettono agli studenti di familiarizzare con le metodologie e tecnologie introdotte, anche investigando soluzioni alternative (applying knowledge and understanding). Inoltre durante le ore di laboratorio è previsto lo sviluppo di un'applicazione realistica con interfaccia web. La preparazione e la discussione del progetto di laboratorio sono inoltre volte a stimolare le capacità di organizzare il lavoro in piccoli gruppi (max 4 studenti), e poi di illustrare verbalmente le

soluzioni adottate (communication skills).

3. Risultati dell'apprendimento attesi:

Lo studente acquisirà la conoscenza delle varie architetture di riferimento per lo sviluppo di applicazioni web, e dei loro pro e contro. Inoltre lo studente acquisirà la conoscenza dei modelli più comunemente adottati per gestire dialoghi mediamente complessi e articolati tra utente e applicazione, e delle tecnologie attualmente utilizzate per l'implementazione delle applicazioni e delle loro interfacce. Lo studente dovrà essere in grado di sviluppare applicazioni Web in ambiente Java, utilizzando JSP e Java Servlet, con interfacce utente basate su HTML5.

4. Modalità di verifica dell'apprendimento:

L'esame è composto da un TEST SCRITTO e da una VERIFICA DI LABORATORIO. Le due prove possono essere sostenute in qualsiasi ordine (cioè, non è necessario aver dato lo scritto per fare la prova di laboratorio, né il vice versa).

Test scritto

Prova scritta che include esercizi e domande teoriche sul programma del corso. Viene valutata da un minimo di 0 ad un massimo di 30 e si considerano sufficienti i voti ≥ 18 . Durante la prova è proibito comunicare con altre persone, presenti in aula o fuori. Inoltre, non si può portare alcun tipo di materiale didattico (appunti, libri, dispense, etc.) ed è vietato usare computer, telefonini o simili. Come da regolamento di Ateneo, ogni studente può sostenere un numero massimo di tre prove scritte durante l'Anno Accademico (cioè, consegnare il proprio elaborato tre volte). Il voto ottenuto durante un test scritto decade se lo studente partecipa ad un altro test scritto e consegna il suo elaborato.

Verifica di laboratorio

Prevede la discussione del progetto di laboratorio svolto durante il corso. La discussione deve essere effettuata preferibilmente in unica soluzione, con tutti i membri del gruppo di laboratorio presenti. Il voto di laboratorio è un numero compreso tra 0 e 30, si considerano sufficienti i voti ≥ 18 .

Calcolo del voto finale dell'esame

Sia X il voto del test scritto; sia Y il voto di laboratorio. Il voto Fin finale dell'esame si ottiene come segue: $Fin = (X+Y)/2$

Note: i voti acquisiti durante la prova di laboratorio, o durante il test scritto, rimangono validi fino al termine della terza sessione d'esame (quella che precede l'inizio del nuovo corso). Quando si superano entrambe le prove, è necessario registrare il voto finale entro i limiti imposti dal Regolamento di Ateneo.

5. Modalità d'insegnamento:

L'insegnamento è diviso in una parte di teoria e una di laboratorio. Per la parte di teoria sono previste 30 ore di lezione frontali, che seguono il programma del corso, integrate da casi di studio e da esercitazioni volte ad illustrare l'applicazione pratica dei concetti appena studiati.

La parte di laboratorio si tiene in altre 30 ore e ha per oggetto l'insegnamento delle tecnologie per lo sviluppo di pagine web statiche e dinamiche (HTML5 e JavaScript) e di applicazioni server-side in ambiente java (JSP e Servlet, con accesso a basi di dati remote attraverso JDBC). Le lezioni si svolgono al computer, in maniera interattiva, e sono corredate da vari esercizi miranti a fornire esempi pratici.

Le lezioni in aula sono svolte principalmente con l'ausilio del computer (proiezione di lucidi). Le esercitazioni in laboratorio sono svolte al computer, utilizzando ambienti di sviluppo professionali, come Eclipse e/o NetBeans.

Le sperimentazioni che vengono effettuate durante le ore di laboratorio, strutturate come sequenze di esercizi specifici, sono fondamentali per aiutare gli studenti a comprendere e assimilare i contenuti teorici spiegati a lezione in quanto permettono di mettere in pratica i concetti e le metodologie illustrate su esempi concreti. Inoltre lo sviluppo del progetto di laboratorio permette di consolidare le conoscenze teoriche in un caso realistico di media complessità.

Si consiglia caldamente la frequenza costante alle lezioni di teoria e di laboratorio. E' inoltre fondamentale che gli studenti si iscrivano al corso online su I-Learn, all'interno del quale il docente mette a disposizione materiale didattico di supporto.

6. Attività di supporto:

Il materiale didattico di supporto (lucidi delle lezioni, link a documentazione, esempi di testo di esami ed altro) è disponibile presso il supporto on-line ai corsi: [qui \(PER ORA PAGINA DEL 2014/15\)](#)

Si noti che i lucidi non sostituiscono il libro di testo, né il materiale integrativo ad esso affiancato.

7. Programma:

Architetture delle applicazioni Web: Web browser e Web server; applicazioni basate su un'architettura a 3 livelli (three tier).

- Il primo livello (client dell'applicazione):
 - Scripting lato client: JavaScript e AJAX.
 - Raccolta dati (via HTML form) e interazione con il web server.
- Il terzo livello (livello dei dati):
 - Accesso a database relazionali: driver ODBC (Open Database Connection); Java Database Connectivity (JDBC).
 - XML: rappresentazione di informazioni (XML Schema e DTD); Manipolazione di documenti XML (XPath).
- Il secondo livello (logica applicativa):
 - Progettazione e sviluppo di applicazioni basate su pagine Web dinamiche (Java Server Pages e Servlet Java) E Java Beans per l'accesso a database.
 - Il Pattern Model View Controller per le applicazioni Web.
 - Progettazione e sviluppo di applicazioni Web basate sul pattern MVC.

8. Testi consigliati e bibliografia:

Libro di testo: Programmazione Web lato Server, di V. Della Mea, L. Di Gaspero, I. Scagnetto, Apogeo, 2007

Altri testi (per gli interessati):

- Programmazione Java - tecniche avanzate, di Deitel e Deitel. Ed. Pearson - Prentice Hall
- Principi di Web design, di Joel Sklar. Ed. Apogeo

Insegnamento**MFN0944 - Laboratorio Avanzato di Basi Dati**

Insegnamento (inglese):	Additional Web Technologies
CFU:	6
Settore:	INF/01 - INFORMATICA
Periodo didattico:	2
Tipologia di Attività Formativa:	B - caratterizzante
Docenti:	Carlo MASERA (Professore a Contratto) Rosa MEO (Titolare)

1. Prerequisiti e Propedeuticità:**Competenze attese in ingresso**

Linguaggi di programmazione ad oggetti e fondamenti di basi di dati.

Eventuali corsi propedeutici

Basi di Dati e Laboratorio, Basi di Dati e Sperimentazione, Basi di Dati e Sistemi Informativi, Modelli Avanzati e Architetture di Basi di Dati.

2. Obiettivi formativi:

Il corso si pone l'obiettivo di studiare nelle sue principali funzionalità uno strumento reale e di ampia diffusione per la gestione delle basi di dati come Oracle. Il corso introduce alle basi di dati di nuova generazione, i NoSQL databases, adatti a gestire grandi moli di dati non strutturati e architetture di gestione distribuita dei dati come Hadoop e Map Reduce (applicazione dette di *big data*). Si propongono esperienze in laboratorio durante le quali si svilupperanno applicazioni di gestione e analisi di grossi moli di dati.

3. Risultati dell'apprendimento attesi:

Esperienza di utilizzo di uno strumento di ampia diffusione come Oracle per la gestione delle basi di dati, sia dal punto di vista utente che dell'amministratore. Conoscenza ed esperienza delle architetture software e di modello dei dati per sviluppare e gestire le applicazioni sui big data.

4. Modalità di verifica dell'apprendimento:

La data dell'esame puo' essere fissata per appuntamento.

Tutti dovranno sostenere:

- breve orale (50% del voto) sul Programma del corso.
- se avete seguito il laboratorio e completato il progetto, potete presentarlo (50% del voto), utilizzando 4-8 lucidi o una piccola documentazione scritta

5. Modalità d'insegnamento:

Le lezioni in aula e in laboratorio sono svolte sia con la proiezione di lucidi animati che con la sperimentazione diretta e lo sviluppo di applicazioni su un sistema di gestione di basi dati.

6. Attività di supporto:

Il materiale didattico di supporto (lucidi, link, esempi ed altro) e' disponibile presso il supporto online ai corsi [Learn](#).

7. Programma:

- Introduzione a Oracle: architettura, modello degli oggetti logico/fisici.
- Il catalogo, creazione del database, le operazioni di start-up e shut-down dell'istanza.
- Dimensionamento degli oggetti, blocco Oracle e clausola di storage, algoritmo di allocazione degli extent.
- Vincoli di integrita' e presentazione dei trigger.
- Esecuzione di un comando SQL.
- Utilities di Import/Export, e Loader.
- Back-up e recovery.
- Introduzione al linguaggio per le stored procedure PL/SQL.
- Introduzione all'utilizzo di SQL dinamico.
- Database distribuiti e replicati.
- NoSql databases: introduzione, modello dei dati, operazioni
- AdHoop e Map reduce: introduzione, modello dei dati, operazioni
- Sviluppo di una piccola applicazione software che coinvolge lo scaricamento, la preparazione e l'analisi di grossi volumi di dati (ad esempio, analisi dei sentimenti nei messaggi degli utenti di una applicazione di social media)

8. Testi consigliati e bibliografia:

1. Rick Greenwald, ORACLE essentials : ORACLE9i, ORACLE8i and ORACLE8 (2 Edizione), O'Reilly, 2001
2. Oracle9i Concepts, Documento Oracle, 2002

(disponibile nella [Documentazione on line](#) per gli studenti)

1. Seven Databases in Seven Weeks: A Guide to Modern Databases and the NoSQL Movement, by Eric Redmond, Jim R. Wilson, Pragmatic Bookshelf, 2012
2. Hadoop: The Definitive Guide, 4th Edition, Storage and Analysis at Internet Scale, by Tom White, O'Reilly Media, 2015

Insegnamento**MFN0962 - Metodi Numerici**

Insegnamento (inglese):	Numerical Methods
CFU:	6
Settore:	MAT/08 - ANALISI NUMERICA
Periodo didattico:	1
Tipologia di Attività Formativa:	C - affine e integrativa
Docenti:	Matteo SEMPLICE (Titolare)

1. Prerequisiti e Propedeuticità:**Competenze attese in ingresso**

Funzioni elementari e loro proprietà. Calcolo differenziale e integrale (inclusa la serie di Taylor). Nozioni di base di algebra lineare (inclusi autovalori ed autovettori). Elementi di programmazione.

Eventuali corsi propedeutici

I corsi della laurea triennale.

2. Obiettivi formativi:

Il corso ha lo scopo di studiare metodologie e tecniche per la soluzione numerica affidabile ed efficiente di problemi e per l'interpretazione consapevole dei risultati. Particolare attenzione viene dedicata all'analisi degli errori e agli aspetti computazionali.

Il corso è completato dall'attività di laboratorio, in cui viene utilizzato il software scientifico MATLAB, per la soluzione di problemi numerici in un ambiente di calcolo scientifico.

3. Risultati dell'apprendimento attesi:

Capacità di progettare e valutare algoritmi numerici e capacità di analizzare criticamente i risultati, con particolare riguardo alla valutazione degli errori.

Conoscenza dei metodi numerici di base nell'ambito dell'algebra lineare, della soluzione di equazioni non lineari, dell'approssimazione di funzioni, dell'integrazione e della soluzione di equazioni differenziali ordinarie.

4. Modalità di verifica dell'apprendimento:

Prova scritta (svolta con l'ausilio del calcolatore) sugli argomenti del corso e dei laboratori.

L'esame orale è facoltativo

5. Modalità d'insegnamento:

Lezioni ed esercitazioni in aula.

Applicazioni in laboratorio informatico con l'uso del software scientifico MATLAB.

6. Attività di supporto:

Il materiale didattico di supporto alle lezioni ed esercitazioni sarà disponibile sul sito di e-learning <http://informatica.i-learn.unito.it...>

7. Programma:

1. Nozioni introduttive.

Rappresentazione dei numeri e sistemi numerici. Aritmetica in virgola mobile, errori di rappresentazione e loro propagazione. Numeri di condizionamento.

2. Algebra lineare numerica.

Richiami di algebra lineare, norme vettoriali e matriciali, numero di condizionamento di una matrice. Metodo di Gauss, fattorizzazione di una matrice. Metodi iterativi per sistemi lineari (Jacobi, Gauss-Seidel). Autovalori e autovettori: teoremi di localizzazione, metodo delle potenze.

3. Interpolazione e approssimazione.

Interpolazione polinomiale. Formule di Lagrange e Newton. Errore di interpolazione, nodi di Chebichev, convergenza. Interpolazione con funzioni polinomiali a tratti (spline). Interpolazione trigonometrica.

Approssimazione. Metodo dei minimi quadrati. Regressione lineare e polinomiale. Metodi di linearizzazione dei dati.

4. Integrazione numerica.

Formule di quadratura di tipo interpolatorio a nodi equidistanti (trapezi, Cavalieri-Simpson, punto medio) e ottimali (gaussiane). Formule di quadratura composte, quadratura automatica.

5. Equazioni non lineari.

Metodi di bisezione, di punto fisso, di Newton e principali varianti. Metodi di globalizzazione (line search). Minimizzazione.

6. Equazioni differenziali ordinarie.

Definizioni ed esempi. Metodi numerici espliciti ed impliciti (Runge-Kutta).

LABORATORIO

Analisi di algoritmi numerici e soluzione numerica di problemi con l'uso del software scientifico MATLAB.

8. Testi consigliati e bibliografia:

Libri consigliati

1. Quarteroni, Sacco, Saleri - Matematica Numerica (available also in English as "Numerical mathematics")
2. V. Comincioli - Metodi numerici e statistici per le scienze applicate - <http://www.multimediacampus.it/offerta/risorse.htm>

Insegnamento**MFN0973 - Modellazione Grafica**

Insegnamento (inglese):	Graphic modeling
CFU:	9
Settore:	INF/01 - INFORMATICA
Periodo didattico:	2
Tipologia di Attività Formativa:	B - caratterizzante
Docenti:	Nello BALOSSINO (Titolare) Davide CAVAGNINO (Titolare) Gianluca FALETTI (Professore a Contratto)

1. Prerequisiti e Propedeuticità:**Competenze attese in ingresso**

Lo studente deve avere una solida conoscenza delle nozioni fornite dai corsi di matematica, con particolare riferimento al calcolo matriciale e alla geometria analitica.

Eventuali corsi propedeutici

Analisi matematica.

2. Obiettivi formativi:

È fornita una conoscenza dettagliata delle tecniche atte a creare forme geometriche bi- e tri-dimensionali e a utilizzare opportuni modificatori morfologici. La costruzione di una scena di sintesi ha infatti come presupposto di base non solo la definizione della geometria degli oggetti che la compongono, ma soprattutto la possibilità di poterne variare l'aspetto a seconda dell'obiettivo e di osservarli da qualsiasi punto di vista. È pertanto necessario acquisire dimestichezza sia con la terminologia sia con le tecniche analitiche che regolano i passi di creazione e modellazione su computer. La sperimentazione con un software (Blender) dei fondamenti teorici appresi fornisce allo studente una visione completa del processo di modellazione.

3. Risultati dell'apprendimento attesi:

Conoscenze delle tecniche di base e avanzate utilizzate per la modellazione di ambienti di sintesi.

4. Modalità di verifica dell'apprendimento:

Progetto in Blender e colloquio orale.

5. Modalità d'insegnamento:

Le lezioni sono di tipo frontale con il coinvolgimento degli studenti in discussioni e approfondimenti a cui si affianca una serie di esercitazioni in laboratorio con utilizzo del software Blender.

6. Attività di supporto:

Appunti delle lezioni forniti dal docente.

7. Programma:

Panoramica sulla computer grafica. Matematica per la computer grafica. Formati di file per la grafica. Equazioni del piano, distanza punto-piano, angolo tra piani, faccia anteriore e posteriore di un poligono. Primitive grafiche di output: sistemi di riferimento, algoritmi per il tracciamento di linee, equazioni lineari, algoritmo DDA, algoritmo di Bresenham per le linee, esempi. Algoritmi per

il tracciamento di circonferenze e ellissi, algoritmo midpoint per circonferenze e ellissi, esempi. Trasformazioni geometriche bidimensionali: traslazione, rotazione, scalamento, rotazione attorno a un punto, scalamento rispetto a un punto. Scalamento generalizzato nelle due direzioni. Ulteriori trasformazioni bidimensionali: riflessioni, taglio. Trasformazioni tra due sistemi di riferimento. Visualizzazione bidimensionale: la pipeline di visualizzazione bidimensionale, finestra di clipping. Trasformazioni di normalizzazione e viewport: mapping delle clipping window nella viewport normalizzata e nel quadrato. Algoritmi di clipping: clipping bidimensionale di punti e linee. Clipping di Cohen-Sutherland Clipping. Clipping di poligoni con algoritmo di Sutherland-Hodgman. Trasformazioni geometriche nello spazio tridimensionale: traslazione tridimensionale, rotazioni tridimensionali, rotazione tridimensionale degli assi coordinati, rotazioni tridimensionali generalizzate. Metodo dei quaternioni per le rotazioni tridimensionali. Scalamento tridimensionale. Composizione di trasformazioni tridimensionali. Altre trasformazioni tridimensionali: riflessioni, taglio. Trasformazioni tra sistemi di assi coordinati tridimensionali. Trasformazioni affini. Visualizzazione tridimensionale: la pipeline di visualizzazione tridimensionale. Parametri di visualizzazione tridimensionale. Trasformazione da coordinate mondiali a coordinate di visualizzazione, trasformazioni di proiezione, proiezioni ortogonali. Proiezioni parallele oblique. Proiezioni prospettiche: trasformazione di coordinate, punti di fuga, volume di vista. Frustum di proiezione prospettica simmetrico. Frustum di proiezione prospettica obliqua. Coordinate normalizzate di trasformazione proiettiva prospettica. Trasformazione di viewport e coordinate schermo tridimensionali. Algoritmi di clipping tridimensionali. Rappresentazione di oggetti tridimensionali: superfici quadratiche, rappresentazione spline. Metodi di interpolazione di spline cubiche: interpolazione di Hermite. Curve spline di Bézier: equazioni delle curve di Bézier, proprietà delle curve di Bézier, curve cubiche di Bézier. Curve B-spline cubiche periodiche. Superfici B-spline. Spline razionali. Quadtree, Octree, BSP-trees. Metodi di geometria frattale: procedure di generazione frattale, classificazione dei frattali, dimensione frattale, costruzione geometrica di frattali deterministici autosimili. Pattern di mezzatinta e tecniche di dithering. Texturing e metodi per il dettaglio delle superfici. Modelli di colore e applicazioni del colore. Bump mapping. Animazione al computer. Progettazione di sequenze di animazione. Sequenze di animazione tradizionali. Sistemi a key-frame. Animazione di soggetti: animazione di figure articolate, motion capture. Movimenti periodici. Esercitazioni con software Blender.

8. Testi consigliati e bibliografia:

- Donald D. Hearn, M. Pauline Baker, Warren Carithers, "Computer Graphics with Open GL (4th Edition)", Pearson Education

- John F. Hughes, Andries van Dam, Morgan McGuire, David F. Sklar, James D. Foley, Steven K. Feiner, Kurt Akeley, "Computer Graphics: Principles and Practice, 3rd Edition", Addison-Wesley

Insegnamento**MFN0940 - Modelli Avanzati e Architetture di Basi di Dati**

Insegnamento (inglese):	Advanced Data Models and Database Architectures
CFU:	9
Settore:	INF/01 - INFORMATICA
Periodo didattico:	1
Tipologia di Attività Formativa:	B - caratterizzante
Docenti:	Giovanni SACCO (Titolare)

1. Prerequisiti e Propedeuticità:**Competenze attese in ingresso**

Conoscenze approfondite del modello relazionale, del modello semantico ER, dello standard SQL2, di sistemi operativi e algoritmi.

Eventuali corsi propedeutici

Le competenze attese in ingresso richieste all'inizio del corso sono fornite da insegnamenti d'area "base di dati", "sistemi operativi" e "Algoritmi" offerti in corsi di laurea di primo livello. Ad esempio: "Basi di Dati e Sperimentazioni", "Algoritmi e sperimentazioni" e "Sistemi operativi e sperimentazioni" della laurea in Informatica dell'Università degli Studi di Torino.

2. Obiettivi formativi:

L'insegnamento e' un corso avanzato su modelli di dati ed architetture dei sistemi databases. I corsi introduttivi sui sistemi database sono basati sul modello relazionale e danno nozioni limitate sull'architettura dei sistemi database. Da un lato, il modello relazionale, anche se e' ormai divenuto uno standard, non e' adatto a contesti in cui i dati sono strutturati ma complessi (ad esempio, sistemi CAD, GIS, ecc.), in cui i dati non sono strutturati (banche dati testuali) o in cui la ricerca che l'utente effettua e' sostanzialmente esplorativa (selezione di prodotti da acquistare, diagnosi medica, ecc.) e non precisa come in ambienti tradizionali. Dall'altro lato, la comprensione dell'architettura interna di un sistema di gestione dati, dalle unita' disco all'ottimizzatore delle queries, e' necessaria per disegnare transazioni ad alta performance in ambiente relazionale, ma anche per realizzare applicazioni ad hoc senza usare un dbms preconfezionato.

Partendo da queste considerazioni, il corso si pone un duplice obiettivo. Per quanto riguarda i modelli, l'obiettivo e' che lo studente sia in grado di affrontare con successo applicazioni in una vasta gamma di ambienti diversi, avendo competenze sui principali modelli per dati strutturati alternativi al modello relazionale (object-oriented e object-relational), sui modelli per database semi-strutturati con ricerca esplorativa (dynamic taxonomies) e non strutturati (information retrieval).

Per quanto riguarda le architetture, l'obiettivo e' che lo studente sia in grado di migliorare la performance per applicazioni critiche e di realizzare sistemi ad hoc, avendo una precisa comprensione dei problemi e delle soluzioni di implementazione ed architetture per sistemi database centralizzati e distribuiti, con particolare riferimento ai sistemi database relazionali e con l'inclusione della gestione dei malfunzionamenti e della concorrenza.

3. Risultati dell'apprendimento attesi:

Gli studenti dovranno avere assimilato le motivazioni, i costrutti e l'uso di modelli object-oriented, object-relational, di information retrieval e le dynamic taxonomies.

Gli studenti acquisiranno competenze approfondite sui metodi di valutazione delle principali operazioni dell'algebra relazionale, sulle principali strutture di accesso, sulle organizzazioni ad indici secondari nell'ambito relazionale e nei motori di ricerca, sull'ottimizzazione delle

interrogazioni, i sistemi di controllo della concorrenza e recupero da malfunzionamenti.

4. Modalità di verifica dell'apprendimento:

La modalità d'esame consiste in una prova scritta, a domande aperte. Il controllo dell'apprendimento durante il corso è basato sull'interazione con gli studenti stimolata durante le ore di lezione, di esercitazione e durante i ricevimenti. Sessioni di domande e risposte alla fine di ogni argomento principale

5. Modalità d'insegnamento:

Lezioni frontali: 90 ore.

Le lezioni in aula sono svolte sia con l'ausilio di lucidi, sia direttamente alla lavagna.

6. Attività di supporto:

Lucidi e articoli aggiuntivi disponibili in Moodle

7. Programma:

Modelli Avanzati

- Il modello ad oggetti e il modello object-relational
- Information Retrieval
- Dynamic Taxonomies e Faceted Search

Architetture dei sistemi di basi di dati

- Dispositivi di memorizzazione (inclusa architettura RAID e dispositivi a stato solido). Ripasso di sistemi operativi: politiche di schedulazione del disco, allocazione dello spazio su disco e file systems.
- Risoluzione di operazioni tramite scansioni sequenziali:
 - selezione, proiezione, join con nested loops, nested scans, merging scans, partizionamento hash ricorsivo
 - Semijoins e filtraggio tramite filtri binari e filtri di Bloom. Altre applicazioni dei filtri di Bloom
- Gestione del buffer
- Indici.
 - Organizzazione con B-trees. B+trees e prefix B-trees. Compressione della chiave. Bufferizzazione
 - Organizzazione tramite hash. Metodi statici. Metodi dinamici: hash estendibile. Tries.
- Indici su chiavi secondarie.
 - Liste multiple e liste invertite Applicazione delle liste invertite a problemi di Information Retrieval (cenni)
- Ottimizzazione delle interrogazioni.
- Distribuzioni non uniformi di accesso: Zipf e 80-20
- Disegno fisico di database
- Transazioni.
 - Gestione dei malfunzionamenti
 - Controllo della concorrenza

8. Testi consigliati e bibliografia:

Libri di testo

- Elmasri, Navathe Sistemi di basi di dati: Complementi, Addison Wesley, 4 edizione, 2005
- Baeza-Yates, Ribeiro-Nieto, Modern Information Retrieval, II edition Addison-Wesley
- Sacco, Tzitzikas, Dynamic taxonomies and faceted search, Springer
- Albano, Costruire sistemi per basi di dati, 2001 oppure Basi di dati: strutture ed algoritmi, 1992
- Elmasri, Navathe Sistemi di basi di dati: Fondamenti, Addison Wesley, 4 edizione, 2004

Articoli

- Sacco, Fragmentation - A technique for efficient query processing, ACM TODS 11:2, 1986
- Sacco, Buffer management in relational database systems, ACM TODS 11:4, 1986
- Comer, The ubiquitous B-tree, ACM CSUR, 11:2, 1979
- Enbody, Du, Dynamic hashing schemes, ACM CSUR, 20:2, 1988

Insegnamento**MFN0991 - Modelli Avanzati e Architetture di Basi di Dati - Parte A**

Insegnamento (inglese):	Advanced Data Models and Database Architectures - A
CFU:	6
Settore:	INF/01 - INFORMATICA
Periodo didattico:	1
Tipologia di Attività Formativa:	D - libera
Docenti:	Giovanni SACCO (Titolare)

1. Prerequisiti e Propedeuticità:**Competenze attese in ingresso**

Conoscenze approfondite del modello relazionale, del modello semantico ER, di sistemi operativi e algoritmi.

Eventuali corsi propedeutici

Le competenze attese in ingresso richieste all'inizio del corso sono fornite da insegnamenti d'area "base di dati", "sistemi operativi" e "Algoritmi" offerti in corsi di laurea di primo livello. Ad esempio: "Basi di Dati e Sperimentazioni", "Algoritmi e sperimentazioni" e "Sistemi operativi e sperimentazioni" della laurea in Informatica dell'Università degli Studi di Torino.

2. Obiettivi formativi:

L'insegnamento è un corso avanzato su architetture dei sistemi database. I corsi introduttivi sui sistemi database danno nozioni limitate sull'architettura dei sistemi database. La comprensione dell'architettura interna di un sistema di gestione dati, dalle unità disco all'ottimizzatore delle queries, è necessaria per disegnare transazioni ad alta performance in ambiente relazionale, ma anche per realizzare applicazioni ad hoc senza usare un dbms preconfezionato. L'obiettivo del corso è che lo studente sia in grado di migliorare la performance per applicazioni critiche e di realizzare sistemi ad hoc, avendo una precisa comprensione dei problemi e delle soluzioni di implementazione ed architetture per sistemi database centralizzati e distribuiti, con particolare riferimento ai sistemi database relazionali e con l'inclusione della gestione dei malfunzionamenti e della concorrenza.

3. Risultati dell'apprendimento attesi:

Gli studenti acquisiranno competenze approfondite sui metodi di valutazione delle principali operazioni dell'algebra relazionale, sulle principali strutture di accesso, sulle organizzazioni ad indici secondari nell'ambito relazionale e nei motori di ricerca, sull'ottimizzazione delle interrogazioni, i sistemi di controllo della concorrenza e recupero da malfunzionamenti.

4. Modalità di verifica dell'apprendimento:

Esame scritto con domande aperte
Il controllo dell'apprendimento durante il corso è basato sull'interazione con gli studenti stimolata durante le ore di lezione, di esercitazione e durante i ricevimenti.
Sessioni di domande e risposte alla fine di ogni argomento principale

5. Modalità d'insegnamento:

Lezioni frontali: 60 ore

Le lezioni in aula sono svolte sia con l'ausilio di lucidi, sia direttamente alla lavagna.

6. Attività di supporto:

Lucidi del corso disponibili in Moodle

7. Programma:

- Dispositivi di memorizzazione (inclusa architettura RAID e dispositivi a stato solido). Ripasso di sistemi operativi: politiche di schedulazione del disco, allocazione dello spazio su disco e file systems.
- Risoluzione di operazioni tramite scansioni sequenziali:
 - selezione, proiezione, join con nested loops, nested scans, merging scans, partizionamento hash recursivo
 - Semijoins e filtraggio tramite filtri binari e filtri di Bloom. Altre applicazioni dei filtri di Bloom
- Gestione del buffer
- Indici.
 - Organizzazione con B-trees. B+trees e prefix B-trees. Compressione della chiave. Bufferizzazione
 - Organizzazione tramite hash. Metodi statici. Metodi dinamici: hash estendibile. Tries.
- Indici su chiavi secondarie.
 - Liste multiple e liste invertite Applicazione delle liste invertite a problemi di Information Retrieval (cenni)
- Ottimizzazione delle interrogazioni.
- Distribuzioni non uniformi di accesso: Zipf e 80-20
- Disegno fisico di database
- Transazioni.
 - Gestione dei malfunzionamenti
 - Controllo della concorrenza

8. Testi consigliati e bibliografia:

Libri di testo

- Albano, Costruire sistemi per basi di dati, 2001 oppure Basi di dati: strutture ed algoritmi, 1992
- Elmasri, Navathe Sistemi di basi di dati: Fondamenti, Addison Wesley, 4 edizione, 2004

Articoli

- Sacco, Fragmentation - A technique for efficient query processing, ACM TODS 11:2, 1986
- Sacco, Buffer management in relational database systems, ACM TODS 11:4, 1986
- Comer, The ubiquitous B-tree, ACM CSUR, 11:2, 1979
- Enbody, Du, Dynamic hashing schemes, ACM CSUR, 20:2, 1988

Insegnamento**MFN0960 - Modelli Concorrenti e Algoritmi distribuiti**

Insegnamento (inglese):	Concurrent Models and Distributed Algorithms
CFU:	6
Settore:	INF/01 - INFORMATICA
Periodo didattico:	1
Tipologia di Attività Formativa:	B - caratterizzante
Docenti:	Ines Maria MARGARIA (Titolare)

1. Prerequisiti e Propedeuticità:**Competenze attese in ingresso**

I prerequisiti per il corso sono la nozioni di base della struttura degli elaboratori, delle reti di calcolatori, dei Sistemi Operativi e una buona familiarità con i linguaggi di programmazione. Sono richieste inoltre conoscenze di matematica discreta, ed elementi di base di analisi degli algoritmi.

Eventuali corsi propedeutici

Non e' richiesto nessun corso particolare della laurea magistrale.

2. Obiettivi formativi:

Gli obiettivi del corso sono sostanzialmente due: 1.fornire le metodologie e gli strumenti di base per la Programmazione Concorrente; 2.introdurre il disegno e l'analisi degli Algoritmi Distribuiti. Per il primo obiettivo l'enfasi è posta non tanto sulla programmazione in uno specifico linguaggio concorrente, quanto sui vari modelli cui i linguaggi concorrenti fanno riferimento. Lo studio di questi modelli riguarda principalmente i costrutti per esprimere la concorrenza dei processi e le loro possibili interazioni. L'analisi viene effettuata esaminando un insieme di esempi che coprono un'area significativa di applicazioni e confrontando varie soluzioni nell'ambito di uno stesso modello e tra i vari modelli. Gli esempi e i relativi codici sono presentati utilizzando uno pseudo-codice C-like. Per quanto riguarda il secondo obiettivo viene presentata una collezione significativa di Algoritmi Distribuiti, rappresentati mediante un modello teorico; l'uso di questo modello permette di formalizzare in maniera adeguata le dimostrazioni di correttezza e l'analisi di complessità. Un ulteriore obiettivo riguarda la capacità di comunicare, a interlocutori specialisti e non specialisti, le conoscenze acquisite in modo chiaro e privo di ambiguità.

3. Risultati dell'apprendimento attesi:

Al termine del corso lo studente dovrà essere in grado di utilizzare in maniera adeguata gli strumenti per esprimere e gestire la concorrenza utilizzando come criteri la modularità, la potenza espressiva, la facilità d'uso, l'affidabilità e l'efficienza. Per quanto riguarda gli algoritmi distribuiti sarà a conoscenza delle problematiche e dei risultati di impossibilità connessi al disegno di tali algoritmi in vari ambienti di rete (sincroni, asincroni, ..), e avrà una buona familiarità con le principali classi di algoritmi distribuiti: mutua esclusione, trattamento del deadlock, elezione del leader, problema del consenso, ecc... Infine lo studente saprà presentare ed esporre una relazione sugli argomenti del corso, utilizzando un linguaggio scientifico adeguato ed efficace.

4. Modalità di verifica dell'apprendimento:

L'esame si articola in una prova scritta seguita da una prova orale. La prova scritta riguarda la scrittura di programmi concorrenti. La prova orale verte sull'intero programma (Programmazione Concorrente + Algoritmi Distribuiti). La parte di Programmazione Concorrente incide sulla votazione finale con un peso di 2/3 mentre la parte sugli Algoritmi Distribuiti ha peso di 1/3.

L'eventuale partecipazione all'attività seminariale prevede una valutazione in trentesimi; la valutazione tiene conto della padronanza dei concetti esposti, della completezza della presentazione, della chiarezza espositiva, dell'efficacia degli esempi forniti, della capacità di stimolare la discussione in classe. Se la votazione è positiva ($> 18/30$) si ottiene l'esonero della parte orale sugli Algoritmi Distribuiti. In questo caso il voto finale viene dato dalla media pesata tra il voto sulla Programmazione Concorrente (scritto + orale) con peso $2/3$ e della votazione del seminario, con peso $1/3$.

5. Modalità d'insegnamento:

La parte relativa alla Programmazione Concorrente (circa 40 ore tra lezioni frontali e esercitazioni) viene svolta in aula, talvolta con l'ausilio di lucidi. Alcune ore del corso sono riservate ad esercitazioni in cui alcuni problemi di concorrenza sono affrontati dagli studenti in piccoli gruppi; le soluzioni proposte sono discusse collegialmente. La parte relativa agli Algoritmi Distribuiti (circa 20 ore di lezione) viene svolta tramite un ciclo di seminari che coinvolge gli studenti. Gli studenti che intendono partecipare all'attività seminariale, approfondiscono un particolare algoritmo, con l'aiuto e la supervisione del docente ne preparano una presentazione con i lucidi e lo espongono ai compagni; all'esposizione segue una discussione con la classe. La partecipazione attiva ai seminari non è obbligatoria, ma per gli studenti che vi aderiscono è richiesta la frequenza all'intero ciclo di seminari. La frequenza alle altre lezioni è facoltativa.

6. Attività di supporto:

Il materiale didattico di supporto sarà messo a disposizione degli studenti su Moodle durante lo svolgimento del corso.

7. Programma:

Concetti fondamentali di Programmazione Concorrente: Processi Concorrenti, Architettura di una macchina concorrente, Costrutti Linguistici per la Programmazione Concorrente. Modello a Memoria Comune: Semafori, Regioni Critiche e Regioni Critiche Condizionali, Monitor, Path expressions. Modello a Memoria Distribuita: Caratteristiche Scambio di Messaggi, Primitive Asincrone, Primitive Sincrone, Chiamate di Procedura Remota, Linguaggi Concorrenti. Coordinazione in Sistemi Distribuiti: Ordinamento degli Eventi, Mutua Esclusione, Atomicità, Controllo della Concorrenza, Gestione del Deadlock, Algoritmi di elezione, Problema del Consenso. Approfondimento di Algoritmi distribuiti: Algoritmi di M.E. e elezione, Algoritmi di gestione del deadlock, Algoritmi di terminazione, Algoritmi di Misra, Algoritmi di consenso.

8. Testi consigliati e bibliografia:

Ancilotti - Boari. Programmazione concorrente e distribuita. McGRAW-HILL Ben-Ari - Principles of Concurrent and Distributed Programming. 2 edition. ADDISON-WESLEY Silberschatz - Galvin. Operating System Principles. ADDISON WESLEY 4 ed. Nancy A. Lynch - Distributed Algorithms - Morgan Kaufmann Raynal. Distributed Algorithms and Protocols. WILEY & SONS

Insegnamento**MFN0953 - Modelli e Metodi per il Supporto alle Decisioni**

Insegnamento (inglese):	Models and Methods for Decision Support Systems
CFU:	6
Settore:	INF/01 - INFORMATICA
Periodo didattico:	2
Tipologia di Attività Formativa:	B - caratterizzante
Docenti:	Roberto ARINGHIERI (Titolare)

1. Prerequisiti e Propedeuticità:**Competenze attese in ingresso**

Lo studente deve avere la capacità di rappresentare in modo astratto problemi computazionali. Lo studente deve avere competenze di analisi e progetto di algoritmi.

Eventuali corsi propedeutici

Calcolo Matriciale e Ricerca Operativa. Algoritmi e Strutture Dati.

2. Obiettivi formativi:

L'insegnamento si propone di fornire allo studente le competenze per la creazione di strumenti di supporto alle decisioni integrando modelli e metodi di analisi what-if (es. simulazione) e what-best (es. ottimizzazione).

3. Risultati dell'apprendimento attesi:

Lo studente deve dimostrare la capacità di affrontare problemi reali in modo da proporre delle soluzioni efficaci ed efficienti utilizzando gli strumenti appresi durante il corso.

4. Modalità di verifica dell'apprendimento:

L'esame è costituito dalla discussione di un progetto, diverso per ogni gruppo di studenti. Un gruppo può essere composto al più da 2 studenti. Entrambi gli studenti devono essere presenti alla discussione. Il voto ottenuto durante una prova rimane valido durante tutto l'Anno Accademico in cui la prova è stata sostenuta. L'iscrizione all'esame è obbligatoria.

5. Modalità d'insegnamento:

Lucidi proiettati in aula e tradizionali spiegazioni alla lavagna. Esercitazioni in laboratorio.

6. Attività di supporto:

Verranno forniti appunti ed altro materiale disponibili alla pagina I-learn del corso.

7. Programma:

1. Introduzione. Richiami ed elementi di base di Simulazione e di Ottimizzazione.
2. Modelli per descrivere workflows: Discrete Event Simulation.
3. Modelli per descrivere l'iterazione tra componenti diverse di un sistema: Agent-based Simulation.
4. Modelli per l'analisi di politiche di gestione di medio e lungo termine: System Dynamics.

5. Algoritmi euristici e metaeuristici di ottimizzazione.

6. Integrazione tra simulazione ed ottimizzazione: case studies.

8. Testi consigliati e bibliografia:

(1) M. C. Fu. Optimization for simulation: Theory vs. practice. *INFORMS Journal on Computing*, 14(3):192-215, 2002.

(2) M.C. Fu, F.W. Glover, and J. April. Simulation optimization: A review, new developments, and applications. In M.E. Kuhl, N.M. Steiger, F.B. Armstrong, and J.A. Joines, editors, *Proceedings of the 2005 Winter Simulation Conference*, pages 83-95, 2005.

(3) P. Terna, R. Boero, M. Morini, and M. Sonnessa. *Simulazione, modelli ad agenti e scienze sociali*. il Mulino, 2006.

(4) J.D. Sterman. *Business Dynamics: Systems Thinking and Modeling for a Complex World*. McGraw-Hill, 2000.

(5) E. Aarts, J. K. Lenstra eds.: "Local search in combinatorial optimization", Wiley, 1997.

(6) Software per la Simulazione: [AnyLogic](#)

(7) Software per l'ottimizzazione: [IBM ILOG CPLEX Optimization Studio](#)

Insegnamento**MFN1349 - Ottimizzazione Combinatoria**

Insegnamento (inglese):	Combinatorial Optimization
CFU:	6
Settore:	MAT/09 - RICERCA OPERATIVA
Periodo didattico:	0
Tipologia di Attività Formativa:	C - affine e integrativa
Docenti:	Roberto ARINGHERI (Titolare) Andrea Cesare GROSSO (Titolare)

1. Prerequisiti e Propedeuticità:**Competenze attese in ingresso**

Per seguire il corso è consigliabile avere chiare alcune nozioni apprese durante il corso di Calcolo Matriciale e Ricerca Operativa e, in particolare, quelle di modelli di programmazione lineare, della teoria della dualità e dell'algoritmo del simplesso

Eventuali corsi propedeutici

Calcolo Matriciale e Ricerca Operativa

2. Obiettivi formativi:

Il corso verte sulla discussione di problemi di ottimizzazione, in particolare di problemi di ottimizzazione combinatoria, con brevi cenni all'ottimizzazione continua. Si pone come obiettivo quello di familiarizzare lo studente con problemi di ottimizzazione che occorrono frequentemente in applicazioni pratiche, permettendogli di riconoscere la difficoltà del problema e fornendogli gli strumenti per risolvere tali problemi. Gli strumenti vanno dagli algoritmi esatti, in grado di restituire sempre la soluzione ottima di un problema, alle euristiche, alle quali ricorrere quando una risoluzione esatta del problema richiederebbe tempi di esecuzione troppo elevati.

3. Risultati dell'apprendimento attesi:

Al termine del corso lo studente dovrebbe essere in grado di riconoscere un problema di ottimizzazione, essere conscio che la prima questione da porsi al riguardo di tale problema è quella di quanto sia difficile il problema, ovvero in quale classe di complessità esso rientri ed infine, almeno per quel che riguarda i problemi di ottimizzazione trattati a lezione, saper costruire i modelli matematici di tali problemi ed essere in grado di applicare le tecniche di risoluzione (esatte, approssimate o euristiche) viste.

4. Modalità di verifica dell'apprendimento:

La verifica si articolerà in una prova scritta dove verranno proposti esercizi del tipo di quelli risolti in aula e domande più legate alla teoria, attraverso cui si cerca di verificare quanto in profondità è andato lo studio da parte dello studente. Viene anche svolta una prova orale con discussione del compito ed eventuali domande di approfondimento.

5. Modalità d'insegnamento:

Le lezioni si tengono in aula con l'ausilio di lucidi e lavagna.

6. Attività di supporto:

Verranno forniti appunti disponibili alla pagina I-learn del corso.

7. Programma:

1. Introduzione alla Ricerca Operativa: dal problema reale agli algoritmi di risoluzione. Esempi di modelli matematici. Richiami di Programmazione Lineare.
2. Problemi PLI: rilassamenti, branch and bound, algoritmi basati su piani di taglio. Applicazione a problemi classici (TSP, zaino).
3. Problemi di flusso su reti: modellazione ed algoritmi di soluzione.
4. Programmazione Dinamica.
5. Algoritmi euristici e ricerca locale.

8. Testi consigliati e bibliografia:

C.H. Papadimitriou, K.R. Steiglitz, "Combinatorial optimization: algorithms and complexity", Prentice Hall (solo per eventuali approfondimenti - reperibile in biblioteca)

Insegnamento**INF0008 - Programmazione per Dispositivi Mobili**

Insegnamento (inglese):	Mobile Device Programming
CFU:	6
Settore:	INF/01 - INFORMATICA
Periodo didattico:	2
Tipologia di Attività Formativa:	B - caratterizzante
Docenti:	Ferruccio DAMIANI (Titolare)

1. Prerequisiti e Propedeuticità:**Competenze attese in ingresso**

Agli studenti è richiesto di padroneggiare i concetti e le nozioni di base presentati nella Laurea in Informatica dell'Università di Torino o in altra laurea che soddisfi i requisiti per ottenere la certificazione GRIN (denominata "Bollino GRIN") . In particolare:

- programmazione (buone abilità di programmazione nel paradigma orientato agli oggetti);
- algoritmi e strutture dati;
- architettura degli elaboratori;
- sistemi operativi;
- basi di dati;
- linguaggi formali e traduttori;
- ingegneria del software;
- interazione uomo-macchina e sviluppo per il web.

2. Obiettivi formativi:

L'insegnamento fornisce una conoscenza di base dei principi di progettazione e dei fondamenti dello sviluppo delle applicazioni mobili, e presenta lo sviluppo su piattaforma Android, su piattaforma iOS, e cross-platform. Viene inoltre presentato il paradigma "aggregate programming" per programmare sistemi distribuiti (come ad es. insiemi di dispositivi mobili) specificando il comportamento globale e derivando automaticamente i comportamenti locali.

3. Risultati dell'apprendimento attesi:

Lo studente acquisirà una conoscenza di base dei principi di progettazione e dei fondamenti dello sviluppo delle applicazioni mobili e possiederà una conoscenza di base del paradigma "aggregate programming". Lo studente sarà in grado di:

- progettare e sviluppare applicazioni mobili su piattaforma Android, su piattaforma iOS, e cross-platform;
- scrivere semplici programmi in un linguaggio che supporti il paradigma "aggregate programming".

4. Modalità di verifica dell'apprendimento:

Gli studenti si organizzeranno in gruppi (costituti di norma da 1 o 2 persone, al massimo da 3). Durante il corso ciascun gruppo negozierà con il docente un progetto di esame (volto alla realizzazione di una applicazione mobile):

1. ogni gruppo dovrà proporre un'idea;
2. industrie e/o docenti potrebbero presentare le loro esigenze
3. la negoziazione dovrà, di norma, avere termine prima della fine del corso e avrà validità per un anno solare a partire dall'inizio del corso.

L'esame verterà sul progetto realizzato:

1. Prima dell'esame (di norma, almeno due settimane prima) il gruppo dovrà consegnare al docente una relazione che illustri il progetto e contenga la documentazione dell'applicazione mobile realizzata.
2. Durante l'esame ogni gruppo dovrà:
 - fare una breve presentazione (circa 20 minuti) per introdurre l'applicazione mobile realizzata e illustrare (motivandole in modo adeguato) le scelte progettuali effettuate,
 - presentare un dimostratore (su un dispositivo reale o su un simulatore),
 - discutere il progetto con il docente (durante la discussione ciascun componente del gruppo dovrà dimostrare di padroneggiare i concetti e le nozioni illustrati dal materiale didattico indicato sul sito moodle del corso), e
 - dimostrare di conoscere le basi del paradigma "aggregate programming" e saper scrivere semplici programmi in un linguaggio che supporti tale paradigma.

Il voto finale sarà stabilito considerando la valutazione di:

- relazione che illustra progetto e presentazione del progetto all'esame---peso 30%;
- implementazione del progetto (completo delle sue parti secondo quanto descritto nella relazione) e presentazione del dimostratore all'esame---peso 30%;
- colloquio (discussione del progetto e del paradigma "aggregate programming")---peso 40%.

L'esame farà riferimento al materiale didattico indicato sul sito moodle dell'ultima edizione del corso.

5. Modalità d'insegnamento:

Sono previste 30 ore di lezione frontali e 30 ore di attività in laboratorio, che seguono il programma riportato più avanti, integrate da casi di studio e da esercitazioni volte ad illustrare l'applicazione pratica dei concetti appena studiati.

6. Attività di supporto:

Il [sito moodle del corso](#) sarà usato per rendere disponibili i riferimenti aggiornati al materiale didattico consigliato e per supportare (attraverso i "forum") la comunicazione e la discussione sugli argomenti del corso.

7. Programma:

1. Introduzione alla progettazione di applicazioni mobili.
2. Sviluppo di applicazioni mobili:
 - su piattaforma Android;
 - su piattaforma iOS;
 - cross-platform.
3. Introduzione al paradigma "aggregate programming".

8. Testi consigliati e bibliografia:

Data la rapida evoluzione della materia oggetto di studio, una parte significativa del materiale contenuto nei testi diventa obsoleto in pochi mesi. Per questo motivo il corso farà uso di slides e riferimenti a materiale on-line e (eventualmente) a testi di recente pubblicazione (che verranno resi disponibili sul sito moodle del corso).

Insegnamento**MFN0954 - Reti Complesse**

Insegnamento (inglese):	Complex Networks
CFU:	6
Settore:	INF/01 - INFORMATICA
Periodo didattico:	2
Tipologia di Attività Formativa:	B - caratterizzante
Docenti:	Giancarlo Francesco RUFFO (Titolare)

1. Prerequisiti e Propedeuticità:**Competenze attese in ingresso**

Una forte conoscenza operativa di nozioni di probabilità e di algebra lineare (a livello di laurea in disciplina scientifica) sarà senz'altro d'aiuto, così come una generale maturità in ambito matematico. La capacità di scrivere codice senza problemi è importante, poiché abilità di programmazione sono richieste per eseguire il progetto finale del corso.

Eventuali corsi propedeutici

Se lo studente proviene da un nostro corso di laurea, deve aver sostenuto i seguenti esami:

MFN0570 - Analisi Matematica

MFN0588 - Calcolo Matriciale e Ricerca Operativa

MFN0600 - Elementi di Probabilità e Statistica

MFN0582 - Programmazione I

MFN0585 - Programmazione II

MFN0597 - Algoritmi e Strutture Dati

MFN0602 - Basi di Dati

MFN0598 - Fisica

Altrimenti, se lo studente proviene da altri corsi di laurea, dovrà verificare se i corsi suddetti hanno un programma equivalente a quelli degli esami da loro superati.

2. Obiettivi formativi:

Questo corso introduce i concetti, i principi e le metodologie principali nel campo interdisciplinare della cosiddetta Scienza delle Reti, con un'attenzione particolare alle tecniche analitiche, alla modellazione e alle applicazioni per il Web e per i Social Media.

Gli argomenti trattati includono lo studio della struttura di una rete (grafo), i modelli matematici delle reti, le topologie delle reti più comuni, la struttura di grafi di grandi dimensioni, le strutture delle comunità, la diffusione epidemica di virus ed informazioni, il PageRank e altre misure di centralità, processi dinamici nelle reti e visualizzazione di grafi.

3. Risultati dell'apprendimento attesi:

Dopo aver superato con successo l'esame di questo corso gli studenti saranno in grado di:

- Definire e calcolare metriche di rete di base

- Descrivere le caratteristiche strutturali di reti tecno-sociali
- Mettere in relazione le proprietà del grafo con le funzioni e l'evoluzione della rete corrispondente
- Mettere in relazioni proprietà locali con l'emergenza di schemi globali
- Esplorare nuove angolazioni per capire i comportamenti collettivi che si osservano in una rete
- Impostare e condurre analisi su dati di reti di grandi dimensioni
- Usare strumenti computazionali per l'analisi di rete, quali la libreria igraph (per R e Python) e GePhi.
- Visualizzare reti per evidenziarne proprietà strutturali e globali

4. Modalità di verifica dell'apprendimento:

Esame orale (60%).

Compito I (20%): relazione scritta (2000-3000 parole).

Compito II (20%): progetti individuali su un'analisi di dati rappresentati in forma di rete (lo sviluppo di codice è normalmente parte del compito).

Per superare l'esame gli studenti devono raggiungere e superare un totale del 60% quando tutti le singoli parti sono state terminate e sommate insieme.

5. Modalità d'insegnamento:

Le lezioni si svolgeranno nella modalità tradizionale frontale, facendo uso sia di lavagna che di diapositive elettroniche, con sperimentazione interattiva degli strumenti computazionali di analisi descritti.

Sono previste testimonianze ed interventi di carattere seminariale di esperti e studiosi anche di altre discipline.

6. Attività di supporto:

Una pagina moodle è stata creata per il corso. Tutto il materiale, così come appunti, note e risorse on line saranno condivise. Usando la piattaforma Moodle, gli studenti saranno in grado di discutere le idee introdotte durante il corso e porre domande al docente e agli altri studenti.

7. Programma:

Scienza delle reti

- Intorduzione alle reti complesse
- Teoria dei grafi e metriche di rete
- Reti casuali
- Reti "Piccolo Mondo"
- Reti ad invarianza di scala
- Reti statiche ed in evoluzione
- Correlazioni di grado
- Comunità
- Fenomeni di diffusione

- Apprendimento e giochi su reti

Casi di studio ed applicazioni

- La struttura del nucleo di Internet - evoluzioni e modellazione
- Struttura del Web - PageRank e reti di documenti
- Reti sociali on line e Social Media - Twitter, Facebook, Amazon, ...
- Visualizzazione di Reti
- Reti di similarità e sistemi di raccomandazione
- Fenomeno del "ricco che si arricchisce"
- Collegamenti, vicini e comunità
- Fenomeni "a cascata" ed epidemici/virali
- Bilanciamento strutturale di una rete
- Analisi del sentimento e delle componenti spaziali e temporali di una rete sociale on line.

8. Testi consigliati e bibliografia:

A.-L. Barabási, Network Science, Cambridge University Press, 2015 (online version: <http://barabasi.com/networksciencebook/>)

D. Easley and J. Kleinberg. Networks, Crowds, and Markets: Reasoning About a Highly Connected World, Cambridge University Press, 2010.

M. E. J. Newman. Networks: An Introduction, Oxford University Press, 2010.

Insegnamento**INF0009 - Reti II**

Insegnamento (inglese):	Computer Networks II
CFU:	6
Settore:	INF/01 - INFORMATICA
Periodo didattico:	1
Tipologia di Attività Formativa:	D - libera
Docenti:	Michele GARETTO (Titolare)

1. Prerequisiti e Propedeuticità:**Competenze attese in ingresso**

Il corso assume conoscenze di base sulle reti di calcolatori come commutazione di pacchetto/circuito, il controllo di errore, di flusso, di congestione, la pila protocollare, indirizzi e instradamento, Ethernet, l'architettura TCP/IP di Internet. Sono inoltre richieste nozioni elementari di analisi matematica, calcolo delle probabilità, processi stocastici.

Eventuali corsi propedeutici

Reti di Calcolatori Complementi di analisi e probabilità

2. Obiettivi formativi:

Negli ultimi due decenni, Internet è passata da strumento di ricerca a una componente fondamentale della società; qualcosa che noi tutti diamo per scontato e usiamo quotidianamente. In questo corso esploreremo perché l'infrastruttura di Internet è stata progettata in questo modo, i suoi principi di base e le scelte architetture. Esamineremo i pro e i contro della architettura attuale, e rifletteremo su come rendere Internet migliore in futuro.

Gli obiettivi del corso sono:

- Acquisire familiarità con lo stato dell'arte nelle reti di calcolatori: architetture, protocolli e sistemi.
- Ottenere una certa pratica nel leggere articoli di ricerca e comprenderli criticamente.
- Imparare a presentare e discutere efficacemente in pubblico un

argomento di reti.

3. Risultati dell'apprendimento attesi:

Nozioni teoriche e metodologiche sul funzionamento e sulla progettazione di reti di calcolatori e di sistemi telematici complessi. Concetti avanzati sulla qualità del servizio. Modelli di base per la simulazione e lo studio di reti di calcolatori. Principali metodologie per il controllo del traffico nelle reti a commutazione di pacchetto. Le principali tecnologie hardware e software ad oggi in uso nelle reti geografiche sia fisse che mobili. Metodologie e tecnologie per l'Ingegneria del Traffico; Caratteristiche avanzate dell'architettura TCP/IP e di Internet con particolare riferimento agli aspetti di routing e di offerta di servizi a qualità del servizio.

4. Modalità di verifica dell'apprendimento:

La modalità di esame preferenziale (per il primo appello) consiste in:

- prova scritta su nozioni fondamentali del corso, nella forma di 5-6 domande a risposta aperta
- presentazione orale a fine corso (tramite slides) su un tema coordinato col docente

Gli appelli successivi prevedono una prova scritta analoga alla precedente e una prova orale.

5. Modalità d'insegnamento:

Proiezione di diapositive (slides) e uso della lavagna tradizionale per sviluppo di modelli analitici e calcoli.

6. Attività di supporto:

-Diapositive usate a lezione in formato pdf. -Articoli e survey presi dalla letteratura e indicati di volta in volta dal docente. -Uso della piattaforma Moodle.

7. Programma:

Argomenti del corso

- Principi fondamentali: architettura end-to-end, nomi e indirizzi,

segnalazione, segmentazione, randomizzazione, indirizione, multiplazione, virtualizzazione, scalabilità.

- Allocazione delle risorse di rete: ingegneria del traffico, controllo di congestione come un problema di allocazione delle risorse, TCP.
- Router design: Code input/output, la classificazione dei pacchetti,

scheduling

- Content Centric Networks: reti di distribuzione di contenuti, nomi vs indirizzi, reti di cache.
- BitTorrent: specifiche del protocollo, modelli prestazionali.
- Reti cellulari: architettura e principi, l'evoluzione degli standard.
- Reti di sensori: principi fondamentali e problematiche principali.
- Reti wireless: complementi su 802.11, Bluetooth, reti ad-hoc, routing e scheduling opportunistici.

8. Testi consigliati e bibliografia:

Reti di calcolatori e internet. Un approccio top-down, 6/Ed. James F. Kurose, Keith W. Ross, Pearson Education Italia, 2013

Communication Networks An Optimization, Control and Stochastic Networks Perspective, R. Srikant, Lei Ying, Cambridge University Press, 2014

Insegnamento**MFN0977 - Reti Neurali**

Insegnamento (inglese):	Neural Networks
CFU:	6
Settore:	INF/01 - INFORMATICA
Periodo didattico:	1
Tipologia di Attività Formativa:	B - caratterizzante
Docenti:	Valentina GLIOZZI (Titolare)

1. Prerequisiti e Propedeuticità:**Competenze attese in ingresso**

Fondamenti di calcolo differenziale, fondamenti di calcolo matriciale. Conoscenza ed uso dei principali linguaggi di programmazione, ed eventualmente dell'ambiente Matlab

Eventuali corsi propedeutici

Analisi Matematica, Calcolo Matriciale e Ricerca Operativa, Programmazione, Sistemi intelligenti.

2. Obiettivi formativi:

Il corso si propone di far acquisire le competenze di base relative ai principali modelli di reti neurali sia dal punto di vista teorico-matematico che dal punto di vista applicativo e del loro utilizzo. A partire dallo studio dei componenti elementari, i neuroni, si arriverà ai principali modelli di reti ed alle più note regole di addestramento.

3. Risultati dell'apprendimento attesi:

Il corso si propone di far acquisire una conoscenza critica dei principali modelli di reti neurali e di consentire il loro utilizzo per la risoluzione di problemi nell'ambito dell'apprendimento automatico e del data mining.

4. Modalità di verifica dell'apprendimento:

L'esame è orale, con verifica delle conoscenze acquisite sia attraverso domande che con la discussione degli algoritmi implementati.

5. Modalità d'insegnamento:

Le lezioni si svolgono sia in aula, dove viene approfondita la parte teorica, che in laboratorio, dove opportune esercitazioni consentiranno di padroneggiare i principali modelli ed algoritmi.

6. Attività di supporto:

Il materiale didattico presentato a lezione (lucidi) è disponibile su ILearn

7. Programma:

Definizione di rete neurale e di neurone. Percettrone, regola delta. Rete neurale a propagazione in avanti, addestramento a retropropagazione dell'errore. Reti neurali a funzioni radiali. Metodo di addestramento ELM. Macchine di Boltzmann e apprendimento stocastico. Reti neurali profonde, caratteristiche e problemi aperti. Apprendimento non supervisionato: modello di Hopfield e suo uso come memoria associativa e per la risoluzione di problemi di ottimizzazione. Self organizing maps, loro struttura e relazione con le tecniche di clustering. Implementazione dei principali modelli in ambiente Matlab

8. Testi consigliati e bibliografia:

S. Haykin - Neural Networks: a Comprehensive Foundation - IEEE Press C. M. Bishop - Pattern recognition and Machine Learning - Springer eds. R. J. Roiger, M. W. Geatz - Introduzione al Data Mining - McGraw-Hill I. H. Witten, E. Frank - Data Mining - 2 ed. - Elsevier eds.

Insegnamento**MFN0945 - Sicurezza I**

Insegnamento (inglese):	Computer and Network Security I
CFU:	6
Settore:	
Periodo didattico:	2
Tipologia di Attività Formativa:	B - caratterizzante
Docenti:	Francesco BERGADANO (Titolare)

1. Prerequisiti e Propedeuticità:**Competenze attese in ingresso**

Si presuppone la conoscenza dei sistemi operativi e delle reti di calcolatori basate sui protocolli della suite TCP/IP. La conoscenza di specifiche piattaforme di sviluppo e programmazione, quali Java e C++, considerata di aiuto alla comprensione degli argomenti svolti nel corso.

Eventuali corsi propedeutici

Sistemi Operativi, Reti di Calcolatori

2. Obiettivi formativi:

Il corso si propone di fornire agli studenti gli strumenti crittografici e tecnici utilizzati per garantire la sicurezza di reti e calcolatori. Inoltre, attraverso l'uso di esempi pratici, il corso fornisce agli studenti una comprensione concreta dei maggiori rischi di sicurezza e delle soluzioni disponibili

3. Risultati dell'apprendimento attesi:

Il corso si propone di preparare gli studenti a lavorare in azienda per la gestione dei sistemi informatici, cooperando con i livelli organizzativi per garantire la sicurezza del sistema informativo nel suo insieme.

4. Modalità di verifica dell'apprendimento:

La verifica viene fatta sulle conoscenze acquisite durante il corso. La durata della prova scritta e' di un'ora e mezza e prevede domande aperte e domande a risposta multipla, relative agli argomenti trattati nel corso

5. Modalità d'insegnamento:

Le lezioni sono di tipo frontale in aula con esempi dal PC del docente collegato in rete e al proiettore durante la lezione.

6. Attività di supporto:

slide online per alcune parti del corso, per il resto si fa riferimento al libro di testo.

7. Programma:

Strumenti crittografici: cifrari simmetrici e asimmetrici, funzioni di hash, firma elettronica
Sicurezza della rete privata: analisi dei rischi di sicurezza informativa, controllo di accesso, protezione da virus, sistemi firewall, reti private virtuali

8. Testi consigliati e bibliografia:

Libro di testo: William Stallings: Cryptography and Network Security Prentice Hall, Seconda Edizione, 1998

Altri Testi per consultazione: Bruce Schneier: Applied Cryptography, John Wiley and sons, 1994.
David Curry: Unix System Security, Addison-Wesley, 1992.

Insegnamento**MFN0952 - Sicurezza II**

Insegnamento (inglese):	Computer and Network Security II
CFU:	6
Settore:	INF/01 - INFORMATICA
Periodo didattico:	1
Tipologia di Attività Formativa:	B - caratterizzante
Docenti:	Francesco BERGADANO (Titolare)

1. Prerequisiti e Propedeuticità:**Competenze attese in ingresso**

Reti, Sistemi Operativi e Crittografia Linguaggi di Programmazione

Eventuali corsi propedeutici

Reti, Sistemi Operativi e Sicurezza del triennio

2. Obiettivi formativi:

L'obiettivo principale del corso e' fornire agli studenti le basi metodologiche e pratiche per l'applicazione degli strumenti gestione e di sicurezza delle reti e dei calcolatori in contesti scientifici, aziendali e professionali.

3. Risultati dell'apprendimento attesi:

Apprendimento di tecniche e di tool specifici per la gestione della sicurezza aziendale. Sistemi di identity management e PKI.

4. Modalità di verifica dell'apprendimento:

Esame orale e discussione progetto di laboratorio

5. Modalità d'insegnamento:

Oltre ad una parte del corso classicamente frontale, sono previsti interventi seminariali da esperti del settore, oltre ad una parte consistente di laboratorio. Tra le risorse aggiuntive di cui si fa richiesta c'e' una borsa ex art. 33 per supporto in laboratorio. Viene inoltre utilizzato Moodle.

6. Attività di supporto:

Documentazione delle attivita' di laboratorio o slide moodle.

7. Programma:

Il corso e' la naturale prosecuzione di quello di "Sicurezza" presente nel triennio. Le differenze tra i due corsi sono sostanziali: il corso della triennale fornisce il background teorico e strutturale sulle vulnerabilita' dei computer e delle reti e sulle soluzioni derivanti dal campo della crittografia applicata. Lo scopo di *Sicurezza II* e' fornire le basi e gli strumenti necessari per affrontare problemi in un contesto reale, dove la gestione dei sistemi e delle reti deve tenere conto di policy interne e di questioni legate all'interoperabilita' tra i diversi ambienti. Il corso e' suddiviso in due parti. La prima parte di carattere teorico-metodologico consta di riferimenti a framework, linguaggi e tecniche studiate in altri corsi, ma inquadrata in un contesto di servizio interoperabile, con particolare riferimento all'identity management e alla sua integrazione in sistemi complessi. La seconda parte del corso (14 ore circa) e' di tipo laboratoriale: e' prevista l'emulazione di ambienti reali e la creazione di testbed nei quali inserire la definizione di utenti e la creazione di certificati di chiave pubblica.

8. Testi consigliati e bibliografia:

1. W. Stallings, *Cryptography and Network Security*, Prentice Hall (versione Italiana: *Crittografia e Sicurezza delle Reti*, Mc Graw Hill Italia)
2. Dispense dei docenti
3. Articoli scientifici e specifiche tecniche messi a disposizione su area WEB (Moodle)

Insegnamento**MFN0949 - Sistemi Cognitivi**

Insegnamento (inglese):	Cognitive Systems
CFU:	9
Settore:	INF/01 - INFORMATICA
Periodo didattico:	2
Tipologia di Attività Formativa:	B - caratterizzante
Docenti:	Guido BOELLA (Titolare) Alessandro MAZZEI (Titolare) Daniele Paolo RADICIONI (Titolare)

1. Prerequisiti e Propedeuticità:**Competenze attese in ingresso**

Si richiedono nozioni di base di Intelligenza Artificiale, in particolare relative all'area di Rappresentazione della Conoscenza. Conoscenze base di algoritmi e di logica formale.

Eventuali corsi propedeutici

Sistemi Intelligenti (propedeuticità consigliata)

2. Obiettivi formativi:

Il corso ha lo scopo di fornire le nozioni di base di Scienze cognitive, sulle ontologie formali e la semantica lessicale, e di approfondire tematiche legate all'elaborazione automatica della lingua.

3. Risultati dell'apprendimento attesi:

- Conoscenza delle basi della Scienza Cognitiva: nascita della Scienza cognitiva, fondamenti metodologici, memoria, conoscenza, emozioni, metafora, classificazione, modelli mentali;
- Conoscenza dell'approccio semantico al Web (Semantic Web) e di alcune tecnologie basate su ontologie formali e loro utilizzo per l'elaborazione automatica del linguaggio;
- Conoscenza di alcune tecniche di elaborazione del linguaggio naturale.

4. Modalità di verifica dell'apprendimento:

Esame orale. Le date sono fissate su appuntamento con i docenti. Per quanto riguarda la parte di corso sulle ontologie (Parte III) la discussione verterà anche sulle esercitazioni.

5. Modalità d'insegnamento:

Lezioni frontali con supporto del videoproiettore e presentazioni PowerPoint.

6. Attività di supporto:

Il corso è supportato da un sito moodle in cui compaiono i lucidi delle lezioni e letture suggerite. Il sito è utilizzato anche per comunicare con gli studenti registrati, ad esempio segnalando seminari o eventi rilevanti.

7. Programma:

Parte I (Prof. Boella) Introduzione alla Scienza cognitiva: nascita della Scienza cognitiva, fondamenti metodologici, memoria, conoscenza, emozioni, metafora, classificazione, modelli mentali;

Parte II (Prof. Mazzei) Introduzione alla linguistica computazionale:
- Linguistica Computazionale Generale

- Introduzione allo studio formale del linguaggio
- Morfologia, Sintassi, Semantica formale
- Parsing e Generazione
- Traduzione Automatica

Parte III (Prof. Radicioni) Introduzione all'uso delle ontologie formali e al loro utilizzo all'interno della linguistica computazionale e del Semantic Web.

INTRODUZIONE ALLE ONTOLOGIE FORMALI E ALLA SEMANTICA LESSICALE Introduzione all'uso delle ontologie formali e al loro utilizzo all'interno della linguistica computazionale e del Semantic Web.

- Knowledge representation e rappresentazioni strutturate
- Il sistema KL-ONE
- WordNet
- Il sistema BabelNet
- FrameNet
- Linguaggi per la modellazione dei dati: XML, RDF e SPARQL - OWL - ConceptNet - Sviluppo di ontologie con Protegé

Questo modulo prevede un numero variabile di esercitazioni, durante le quali saranno proposte problemi legati agli argomenti trattati a lezione.

8. Testi consigliati e bibliografia:

Introduzione alla Scienza cognitiva:

"Scienza Cognitiva" di Bruno Bara, editore Bollati Boringhieri Capitoli: 2, 3.1-3.2, 3.3-3.5, 4, 5, 7.3

La teoria contemporanea della metafora [http://terpconnect.umd.edu/~israel/lakoff-](http://terpconnect.umd.edu/~israel/lakoff-ConTheorMetaphor.pdf)

ConTheorMetaphor.pdf, "Women, fire and dangerous things" di George Lakoff e Mark Johnson (parte iniziale su argomenti visti a lezione) "Mental Models" di P. Johnson-Laird Capitoli 2, 4 e 5 (in alternativa a capitolo 7.3 del libro di Bara)

I lucidi che verranno pubblicati durante il corso si trovano sul sito i-learn assieme ad altre letture (articoli scientifici)

Per la parte di linguistica computazionale:

Daniel Jurafsky and James H. Martin, "An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition", Second Edition, Prentice Hall, 2009. Chapters 19, 20, 23.1.

Insegnamento**MFN0795 - Sistemi di Calcolo Paralleli e Distribuiti**

Insegnamento (inglese):	Parallel and Distributed Computer Systems
CFU:	6
Settore:	INF/01 - INFORMATICA
Periodo didattico:	2
Tipologia di Attività Formativa:	B - caratterizzante
Docenti:	Marco ALDINUCCI (Titolare)

1. Prerequisiti e Propedeuticità:**Competenze attese in ingresso**

Agli studenti sono richieste le seguenti conoscenze di base:

- Architettura degli elaboratori, programmazione imperativa, conoscenza dei linguaggi C/C++ (anche elementare), algoritmi.
- Uso di sistemi UNIX mediante shell, editor, compilazione di programmi (C/C++, Java).
- Conoscenza della lingua inglese (almeno sufficiente per capire testi e manuali in inglese).

Eventuali corsi propedeutici

Costituiscono prerequisiti i contenuti dei corsi di Algoritmi, Programmazione I e II, Architetture degli elaboratori I

2. Obiettivi formativi:

Il corso introduce i principali modelli e strumenti per la programmazione parallela e distribuita, con maggiore enfasi sulla programmazione parallela. Obiettivo primario è fornire metodi e strumenti per dominare la complessità della progettazione di applicazioni parallele basate sui modelli di programmazione a memoria condivisa, a scambio di messaggi e SIMD.

Metodologicamente, il corso prima introduce le architetture ed i concetti fondamentali per la programmazione parallela e distribuita (attività, sincronizzazione, comunicazione), che poi vengono applicati ed esemplificati su esempi di interesse per il curriculum dello studente (es. calcolo scientifico).

3. Risultati dell'apprendimento attesi:

- Conoscenza dei paradigmi di programmazione parallela.
- Competenze di programmazione parallela con i tradizionali modelli di programmazione a basso livello di astrazione: message passing e shared memory.
- Conoscenza degli strumenti di studio delle prestazioni di programmi paralleli.
- Competenze di architetture dei sistemi paralleli multi-core e distribuiti (livello elementare).
- Competenze di programmazione GPGPU (livello elementare).
- Competenze di performance tuning (livello elementare).

4. Modalità di verifica dell'apprendimento:

L'esame è costituito da un progetto e una prova orale, che verte su tutto il programma svolto.

Il progetto consiste nella progettazione, l'implementazione e l'analisi di un software parallelo mediante gli strumenti presentati nel corso. La scelta del software da realizzare è proposta dallo studente e concordata con il docente. Il progetto può essere realizzato in gruppo o individualmente e deve essere accompagnato da una relazione scritta. La relazione deve riportare le scelte progettuali, le sperimentazioni e l'analisi dei risultati ottenuti. L'ammissione alla prova

orale è condizionata al conseguimento di un punteggio di almeno 18/30.

La prova orale - sempre individuale - consiste in un colloquio sugli argomenti svolti durante il corso e prevede, in particolare:

- l'eventuale discussione del progetto e della relazione - l'eventuale svolgimento di esercizi; - l'esposizione di argomenti e risultati trattati nel corso, incluse alcune dimostrazioni.

Entrambe le prove devono essere superate nella stessa sessione d'esame.

5. Modalità d'insegnamento:

Corso mutuato dal CdS in Fisica, *tenuto presso il Dipartimento di Fisica*. Pagina web corrispondente: <http://fisica.campusnet.unito.it/do...>

Corso tradizionale con esercitazioni tenute dal docente con utilizzo alternato di lavagna tradizionale e proiezione di slide. Registro delle lezioni sul sito moodle del corso (inclusivo del materiale didattico ed esercizi proposti).

6. Attività di supporto:

Barry Wilkinson, Michael Allen Parallel Programming: Techniques and Applications Using Networked Workstations and Parallel Computers Prentice Hall; 2 edition ISBN-10: 0131405632

Marco Danelutto DISTRIBUTED SYSTEMS: PARADIGMS AND MODELS, 2011 (Dispensa)

7. Programma:

1) Introduzione alle architetture parallele e distribuite (8 ore)

- architetture a memoria condivisa: SMP, multi-core (4 ore)
- distribuita: MPP, cluster, grid (4 ore)

2) Paradigmi di programmazione (18 ore)

- a basso livello di astrazione: scambio di messaggi, memoria condivisa (8 ore)
- esempi di parallelizzazione di problemi classici (8 ore)
- ad alto livello di astrazione (cenni): componenti, servizi, workflow, skeleton (2 ore)

3) Esempi di uso ed esercitazioni (22 ore)

- Programmazione con thread_POSIX e MPI, esercizi su casi di studio (12 ore)
- Programmazione SIMT di GPGPU (10 ore)

8. Testi consigliati e bibliografia:

Testi consigliati e bibliografia

- Barry Wilkinson, Michael Allen Parallel Programming: Techniques and Applications Using Networked Workstations and Parallel Computers Prentice Hall; 2 edition ISBN-10: 0131405632
- Marco Danelutto DISTRIBUTED SYSTEMS: PARADIGMS AND MODELS, 2011 (Dispensa)
- Articoli scientifici e siti web di approfondimento saranno forniti durante il corso mediante la piattaforma Moodle.

Insegnamento**MFN0978 - Sistemi di Realtà Virtuale**

Insegnamento (inglese):	Virtual Reality Systems
CFU:	9
Settore:	INF/01 - INFORMATICA
Periodo didattico:	1
Tipologia di Attività Formativa:	B - caratterizzante
Docenti:	Nello BALOSSINO (Titolare) Marco GRANGETTO (Titolare) Maurizio LUCENTEFORTE (Titolare)

1. Prerequisiti e Propedeuticità:**Competenze attese in ingresso**

Buona conoscenza delle tecniche di analisi e di calcolo matriciale.

2. Obiettivi formativi:

Gli scopi fondamentali di questo insegnamento consistono sia nell'armonizzare i contributi interdisciplinari che concorrono alla creazione di ambienti virtuali sia nel fornire gli strumenti idonei alla renderizzazione in real time.

3. Risultati dell'apprendimento attesi:

Acquisizione delle conoscenze fondamentali per la navigazione di ambienti di sintesi mediante l'utilizzo di motori di realtà virtuale.

4. Modalità di verifica dell'apprendimento:

La prova d'esame comprende un progetto di laboratorio ed un colloquio orale.

5. Modalità d'insegnamento:

Le lezioni prevedono discussioni in aula ed esercitazioni. Ad integrazione delle lezioni potranno essere tenuti seminari da esperti del settore. Per le esercitazioni sono inoltre utilizzati applicativi appositi (es. Matlab, Unity3D).

6. Attività di supporto:

Il materiale didattico di supporto è disponibile presso il supporto on-line ai corsi [I-learn](#).

7. Programma:**La Realtà Virtuale**

- Una visione d'insieme
- Definizioni, problemi e soluzioni
- Applicazioni

La pipeline di rendering

- Architettura generale
- L'Application Stage
- Il Geometry Stage
- Il Rasterizer Stage

Matematica per gli ambienti virtuali

- Punti e vettori
- Operazioni con in vettori: addizione e sottrazione, norma, prodotto scalare e vettoriale
- Equazione vettoriale del piano
- Direzioni e angoli
- Rappresentazioni matriciali delle trasformazioni
- Quaternioni: definizioni e operazioni

Rappresentazione visuale

- Sorgenti luminose
- Materiali
- Ombreggiature

Illuminazione globale

- Ray tracing
- Radiosity

Costruzione di una scena 3D

- Spazio di vista
- Culling
- Clipping
- Rimozione delle facce nascoste: approccio object space e image space

Recupero della terza dimensione

- La prospettiva
- Ricostruzione delle forme dalle ombre
- Stereovisione
- Determinazione dei punti coniugati
- Cenni su formati 3D e tecniche di compressione

Animazione-simulazione

- Strutture articolate: cinematica diretta, cinematica inversa,
- Corpi rigidi: dinamica, urto elastico ed anelastico, attrito statico e dinamico
- Fenomeni naturali

Intersezioni e collisioni

- Richiami di geometria analitica
- Tipi di bounding-box
- Intersezione retta-piano, retta-sfera, retta-poligono, retta-poliedro, poliedro-poliedro
- Intersezione di bounding-box
- Tecniche di pruning: bounding volume, space partition, hierarchical bounding volume.

Audio 3D

- Stereo e surround
- HRTF (Head-Related Transfer Function)

Laboratorio di Realtà Virtuale

- Introduzione allo sviluppo di applicazioni 3D realtime con Unity3D.

8. Testi consigliati e bibliografia:

- Appunti delle lezioni.
- Tomas Akenine-Moller, Eric Haines, Real time rendering, Third Edition, AK Peters, 2008.
- William R.Sherman, Alan B. Craig, Understanding Virtual Reality, Interface, Application, and Design, Morgan Kaufmann Publisher, 2003.

- Mel Slater, Anthony Steed Yiorgos Chrysanthou, Computer Graphics and virtual environment, Addison Wesley, 2002.

Insegnamento**MFN0941 - Sviluppo Software per Componenti e Servizi Web**

Insegnamento (inglese):	Component Software and Web Services Development
CFU:	9
Settore:	INF/01 - INFORMATICA
Periodo didattico:	1
Tipologia di Attività Formativa:	B - caratterizzante
Docenti:	Giovanna PETRONE (Titolare)

1. Prerequisiti e Propedeuticità:**Competenze attese in ingresso**

Lo studente deve preliminarmente avere la capacità di scrivere, compilare e verificare la correttezza di programmi in Java, la capacità di disegnare interfacce grafiche uomo-macchina utilizzando una delle varie tecnologie web (fornite nei corsi di Programmazione III, Servizi Web e Tecnologie Web), avere conoscenze di base del linguaggio UML (fornite nel corso Sviluppo Applicazioni Software)

Eventuali corsi propedeutici

Laboratorio di Servizi Web, Tecnologie Web

2. Obiettivi formativi:

Il corso ha come obiettivo di fornire agli studenti una panoramica delle tecnologie di sviluppo di sistemi mission critical con particolare enfasi alle soluzioni industriali per l'e-Business, incluso cenni di Cloud Computing. Fornire agli studenti metodologie e strumenti di Project Management per permettere di gestire lo sviluppo di progetti Software dall'analisi al testing. Le basi del Project Management unite all'applicazione della metodologie Agile di Ingegneria del Software si applicheranno allo sviluppo del progetto svolto in laboratorio. L'insegnamento permette agli studenti di familiarizzare con terminologie di uso comune nell'industria moderna quali: sviluppo per componenti, architetture multilivello, middleware. Per la parte di laboratorio viene realizzata un'applicazione completa basata su: modello UML, architettura three-tier, modello a componenti EJB e programmazione in Java e applicazioni Android. Particolare attenzione all'integrazione con ambienti Cloud e le reti sociali, come esempio di integrazione di servizi eterogenei. I temi introdotti durante il progetto di laboratorio corredano e integrano le conoscenze derivanti dalla parte teorica (knowledge and understanding), al tempo stesso presentando problemi realistici di costruzione di un'applicazione complessa, anche investigando soluzioni alternative (applying knowledge and understanding). La preparazione, le presentazioni intermedie e la discussione del progetto sono volte a stimolare le capacità di organizzare il lavoro in piccoli gruppo (2-3 studenti), e poi di illustrare verbalmente le soluzioni adottate (communication skills).

3. Risultati dell'apprendimento attesi:

Lo studente acquisirà la conoscenza delle architetture software, e dovrà essere in grado di ragionare sulle architetture Web per sistemi aziendali di grandi dimensioni, basandosi sull'esperienza del progetto di laboratorio. Lo studente sarà in grado di partecipare allo sviluppo di un progetto software, utilizzando anche le conoscenze relative alle metodologie di sviluppo Agile e anche utilizzando elementi di Project Management agile. Inoltre lo studente sarà in grado di progettare e realizzare lo sviluppo di un'applicazione Java Enterprise.

4. Modalità di verifica dell'apprendimento:

L'esame consiste:

Ai fini della determinazione del voto finale l'esame dell'insegnamento di è diviso in 2 parti.

1. Discussione sul progetto finale di laboratorio obbligatoria. Questa presentazione viene fatta in gruppo e vale il 50% del voto. 2. Orale individuale obbligatoria sulla parte più teorica del corso, serve anche ad individuare eventuali lacune non rilevate nella presentazione del progetto di laboratorio e si ottiene l'altro 50% del voto.

Per la parte di laboratorio, vengono svolti in laboratorio, esercizi che permettono di completare il progetto finale componendo le tecnologie sviluppate negli esercizi. Questo si combina bene con la metodologia dell'Extreme Programming che suggerisce lo svolgimento di Spikes per fornire una pianificazione più corretta del progetto. Queste esercitazioni favoriscono l'assimilazione graduale degli argomenti introdotti durante il corso. Inoltre vengono richieste 2 presentazioni intermedie, che hanno lo scopo di fornire Milestones del progetto e di permettere agli studenti di esercitare la presentazione di un lavoro, oltre che di applicare metodologie di Project Management come le Project Review.

L'esercitazione finale generalmente consiste in un progetto di maggiore impegno, e richiede di sviluppare un'applicazione in cui lo studente deve mostrare di padroneggiare i concetti e gli strumenti presentati durante il laboratorio e gli studenti dovranno presentare i risultati utilizzando uno schema di presentazione che viene utilizzato durante le presentazioni intermedie.

5. Modalità d'insegnamento:

L'insegnamento è diviso in una parte di teoria e una di laboratorio, molto connesse tra loro.

Per la parte di teoria sono previste 50 ore di lezione frontali che seguono il programma riportato più avanti, integrate da casi di studio e da esercitazioni volte ad illustrare l'applicazione pratica dei concetti appena studiati.

La parte di laboratorio è consistita in un modulo da 40 ore, nel quale gli studenti svolgeranno un progetto di laboratorio, a partire dalla proposta con relativa presentazione, fino alla realizzazione completa dell'applicazione, mostrando la demo funzionante, corredata dall'esposizione delle fasi di progettazione. Le lezioni si svolgono in maniera interattiva e sono corredate da vari esercizi miranti a fornire esempi pratici.

6. Attività di supporto:

Il materiale è disponibile su Moodle. <http://informatica.i-learn.unito.it/course/view.php?id=564>

7. Programma:

PARTE DI TEORIA:

Project management cenni. In particolare come gestire un progetto Software, gestire un gruppo di sviluppatori e applicare le metodologie di ingegneria del Software XP.

Le diverse architetture per la progettazione di applicazioni client/server distribuite sulla rete, dalle soluzioni two-tiers alle three-tiers, tipologie varie di middleware e di oggetti distribuiti: i business objects. La distribuzione dei processi applicativi nelle reti Internet ed Intranet. * L'architettura per oggetti distribuiti dello standard CORBA: l'IDL, il linguaggio per la definizione delle interfacce, e gli ORB, gli Object Request Broker. La soluzione DCOM, OLE 2 e ActiveX della Microsoft. L'architettura Java 2 Enterprise: gli Enterprise Java Beans. Nuove soluzioni per le applicazioni Internet ed il commercio elettronico: Java Server Pages per le interfacce utente e XML per l'interscambio di dati. I Web Services e gli standard UML. Cenni di soluzioni per Enterprise Application Integration e le Service Oriented Architecture il Cloud Computing Configuration management e controllo di versione

PARTE DI LABORATORIO:

Definizione e successiva realizzazione dell'applicazione distribuita in Java ed EJB, utilizzando l'application server di Sun J2EE e Netbeans o Eclipse, ambienti di sviluppo per J2EE. Si utilizzerà UML (Unified Modeling Language) nel progetto preliminare di un'applicazione distribuita. Si costruiranno Web Services sia SOAP che RESTful. Inoltre l'applicazione dovrà integrare servizi

offerti dagli Open API piu' noti, come Facebook, Google e si sviluppera' una parte dell'interfaccia per Smartphone nuova generazione (Android) Utilizzo del sistema di controllo versione GIT e della piattaforma Cloud, GAE.

8. Testi consigliati e bibliografia:

G. Alonso and F. Casati and H. Kuno and V. Machiraju, Web services - Concepts, architectures and applications, Springer 2004

JAVA - ENTERPRISE EDITION 5 DIOTALEVI FILIPPO Progettazione e sviluppo di applicazioni Web
HOEPLI Collana: Manuali Hoepli.it Pubblicazione: 06/2006

<http://java.sun.com/blueprints/patterns/>

<http://www.netbeans.org/index.html>

Insegnamento**MFN0995 - Sviluppo Software per Componenti e Servizi Web - Parte A**

Insegnamento (inglese):	Component Software and Web Services Development - A
CFU:	6
Settore:	INF/01 - INFORMATICA
Periodo didattico:	1
Tipologia di Attività Formativa:	B - caratterizzante
Docenti:	Giovanna PETRONE (Titolare)

1. Prerequisiti e Propedeuticità:**Competenze attese in ingresso**

Lo studente deve preliminarmente avere la capacità di scrivere, compilare e verificare la correttezza di programmi in Java, la capacità di disegnare interfacce grafiche uomo-macchina utilizzando una delle varie tecnologie web (fornite nei corsi di Programmazione III, Servizi Web e Tecnologie Web), avere conoscenze di base del linguaggio UML (fornite nel corso Sviluppo Applicazioni Software)

Eventuali corsi propedeutici

Laboratorio di Servizi Web, Tecnologie Web

2. Obiettivi formativi:

Il corso ha come obiettivo di fornire agli studenti una panoramica delle tecnologie di sviluppo di sistemi mission critical con particolare enfasi alle soluzioni industriali per l'e-Business, incluso cenni di Cloud Computing. Fornire agli studenti metodologie e strumenti di Project Management per permettere di gestire lo sviluppo di progetti Software dall'analisi al testing. Le basi del Project Management unite all'applicazione della metodologie Agile di Ingegneria del Software si applicheranno allo sviluppo del progetto svolto in laboratorio. L'insegnamento permette agli studenti di familiarizzare con terminologie di uso comune nell'industria moderna quali: sviluppo per componenti, architetture multilivello, middleware. Per la parte di laboratorio viene realizzata un'applicazione completa basata su: modello UML, architettura three-tier, modello a componenti EJB e programmazione in Java e applicazioni Android. Particolare attenzione all'integrazione con ambienti Cloud e le reti sociali, come esempio di integrazione di servizi eterogenei. I temi introdotti durante il progetto di laboratorio corredano e integrano le conoscenze derivanti dalla parte teorica (knowledge and understanding), al tempo stesso presentando problemi realistici di costruzione di un'applicazione complessa, anche investigando soluzioni alternative (applying knowledge and understanding). La preparazione, le presentazioni intermedie e la discussione del progetto sono volte a stimolare le capacità di organizzare il lavoro in piccoli gruppo (2-3 studenti), e poi di illustrare verbalmente le soluzioni adottate (communication skills).

3. Risultati dell'apprendimento attesi:

Lo studente acquisirà la conoscenza delle architetture software, e dovrà essere in grado di ragionare sulle architetture Web per sistemi aziendali di grandi dimensioni, basandosi sull'esperienza del progetto di laboratorio. Lo studente sarà in grado di partecipare allo sviluppo di un progetto software, utilizzando anche le conoscenze relative alle metodologie di sviluppo Agile e anche utilizzando elementi di Project Management agile. Inoltre lo studente sarà in grado di progettare e realizzare lo sviluppo di un'applicazione Java Enterprise.

4. Modalità di verifica dell'apprendimento:

L'esame consiste:

Ai fini della determinazione del voto finale l'esame dell'insegnamento di è diviso in 2 parti.

1. Discussione sul progetto finale di laboratorio obbligatoria. Questa presentazione viene fatta in gruppo e vale il 50% del voto. 2. Orale individuale obbligatoria sulla parte più teorica del corso, serve anche ad individuare eventuali lacune non rilevate nella presentazione del progetto di laboratorio e si ottiene l'altro 50% del voto.

Per la parte di laboratorio, vengono svolti in laboratorio, esercizi che permettono di completare il progetto finale componendo le tecnologie sviluppate negli esercizi. Questo si combina bene con la metodologia dell'Extreme Programming che suggerisce lo svolgimento di Spikes per fornire una pianificazione più corretta del progetto. Queste esercitazioni favoriscono l'assimilazione graduale degli argomenti introdotti durante il corso. Inoltre vengono richieste 2 presentazioni intermedie, che hanno lo scopo di fornire Milestones del progetto e di permettere agli studenti di esercitare la presentazione di un lavoro, oltre che di applicare metodologie di Project Management come le Project Review.

L'esercitazione finale generalmente consiste in un progetto di maggiore impegno, e richiede di sviluppare un'applicazione in cui lo studente deve mostrare di padroneggiare i concetti e gli strumenti presentati durante il laboratorio e gli studenti dovranno presentare i risultati utilizzando uno schema di presentazione che viene utilizzato durante le presentazioni intermedie.

5. Modalità d'insegnamento:

L'insegnamento è diviso in una parte di teoria e una di laboratorio, molto connesse tra loro.

Per la parte di teoria sono previste 50 ore di lezione frontali che seguono il programma riportato più avanti, integrate da casi di studio e da esercitazioni volte ad illustrare l'applicazione pratica dei concetti appena studiati.

La parte di laboratorio è consista in un modulo da 40 ore, nel quale gli studenti svolgeranno un progetto di laboratorio, a partire dalla proposta con relativa presentazione, fino alla realizzazione completa dell'applicazione, mostrando la demo funzionante, corredata dall'esposizione delle fasi di progettazione. Le lezioni si svolgono in maniera interattiva e sono corredate da vari esercizi miranti a fornire esempi pratici.

6. Attività di supporto:

Il materiale è disponibile su Moodle. <http://informatica.i-learn.unito.it/course/view.php?id=564>

7. Programma:

PARTE DI TEORIA:

Le diverse architetture per la progettazione di applicazioni client/server distribuite sulla rete, dalle soluzioni two-tiers alle three-tiers, tipologie varie di middleware e di oggetti distribuiti: i business objects. La distribuzione dei processi applicativi nelle reti Internet ed Intranet. * L'architettura per oggetti distribuiti dello standard CORBA: l'IDL, il linguaggio per la definizione delle interfacce, e gli ORB, gli Object Request Broker. La soluzione DCOM, OLE 2 e ActiveX della Microsoft. L'architettura Java 2 Enterprise: gli Enterprise Java Beans. Nuove soluzioni per le applicazioni Internet ed il commercio elettronico: Java Server Pages per le interfacce utente e XML per l'interscambio di dati. I Web Services e gli standard UML. Cenni di soluzioni per Enterprise Application Integration e le Service Oriented Architecture il Cloud Computing Configuration management e controllo di versione

PARTE DI LABORATORIO:

Definizione e successiva realizzazione dell'applicazione distribuita in Java ed EJB, utilizzando l'application server di Sun J2EE e Netbeans o Eclipse, ambienti di sviluppo per J2EE. Si utilizzerà UML (Unified Modeling Language) nel progetto preliminare di un'applicazione distribuita. Si costruiranno Web Services sia SOAP che RESTful. Inoltre l'applicazione dovrà integrare servizi offerti dagli Open API più noti, come Facebook, Google e si svilupperà una parte dell'interfaccia per Smartphone nuova generazione (Android) Utilizzo del sistema di controllo versione GIT.

8. Testi consigliati e bibliografia:

G. Alonso and F. Casati and H. Kuno and V. Machiraju, Web services - Concepts, architectures and applications, Springer 2004

JAVA - ENTERPRISE EDITION 5 DIOTALEVI FILIPPO Progettazione e sviluppo di applicazioni Web
HOEPLI Collana: Manuali Hoepli.it Pubblicazione: 06/2006

<http://java.sun.com/blueprints/patterns/>

<http://www.netbeans.org/index.html>

Insegnamento**MFN0961 - Teoria dell'Informazione**

Insegnamento (inglese):	Information theory
CFU:	9
Settore:	INF/01 - INFORMATICA
Periodo didattico:	2
Tipologia di Attività Formativa:	B - caratterizzante
Docenti:	Marco GRANGETTO (Titolare) Matteo SERENO (Titolare)

1. Prerequisiti e Propedeuticità:**Competenze attese in ingresso**

L'organizzazione del modulo presuppone un buona conoscenza delle nozioni fornite dai corsi di probabilità e statistica e più in generale dai corsi di matematica.

2. Obiettivi formativi:

Il corso fornisce gli elementi essenziali della teoria dell'informazione che permettono di comprendere, analizzare e progettare i sistemi per la codifica e la trasmissione dell'informazione. Le competenze acquisite dagli studenti saranno la conoscenza della teoria di Shannon e delle sue molteplici applicazioni in alcuni settori dell'informatica.

3. Risultati dell'apprendimento attesi:

Al termine del corso lo studente avrà acquisito i concetti fondamentali della teoria dell'informazione e sarà in grado di applicarne i risultati alla comprensione e al progetto di sistemi di codifica e trasmissione dell'informazione.

4. Modalità di verifica dell'apprendimento:

La verifica si articolerà in una prova scritta e in un esame orale. Le due parti della verifica potranno richiedere complessivamente un paio d'ore. La verifica sarà orientata ad accertare la capacità critica maturata dallo studente nella valutazione dell'applicabilità a casi di studio reali dei principi teorici studiati nelle prime tre parti costituenti gli argomenti del corso.

5. Modalità d'insegnamento:

Le lezioni in aula si svolgono in maniera tradizionale (lavagna e gesso) e, ove la natura degli argomenti esposti lo consenta, con l'ausilio del calcolatore. Le esercitazioni saranno in parte basate sulla soluzione di esercizi propedeutici alla prova d'esame e in parte dedicate allo sviluppo di esperienze di laboratorio.

6. Attività di supporto:

Il materiale utilizzato dai docenti durante il corso (lucidi, esercizi, articoli di approfondimento) verrà sempre reso disponibile attraverso le pagine web del corso.

7. Programma:

Il corso è strutturato in tre parti.

La prima parte del corso è dedicata alla teoria dell'informazione classica: definizione dell'informazione e tipi di sorgente, concetto di entropia, la codifica di sorgente, primo teorema di Shannon (o della codifica di sorgente), codici univocamente decodificabili, ottimalità della codifica di Huffman, modelli di canale rumoroso, definizione della capacità di canale, secondo teorema di Shannon (o della codifica di canale).

La seconda parte del corso è dedicata allo studio di schemi di codifica di sorgente e canale utilizzati in molteplici applicazioni e sistemi di comunicazione. Per quanto riguarda la codifica di sorgente saranno analizzati la codifica aritmetica, la codifica di Lempel-Ziv-Welch e alcuni standard per la compressione di immagini e video. Tra gli schemi di codifica di canale verranno studiati i codici a blocco lineari, i codici ciclici e i codici convoluzionali.

La terza parte del corso è dedicata all'approfondimento di alcune tematiche, anche attraverso attività sperimentali e di laboratorio. I temi trattati verranno scelti in base alla loro rilevanza nel campo della ricerca scientifica oppure per la loro diffusione in sistemi commerciali (GSM,UMTS, DVB, WiFi).

8. Testi consigliati e bibliografia:

- Thomas M. Cover, Joy A. Thomas, Elements of Information Theory, Wiley, 1991 , 2006(2nd edition).
- Raymond W. Yeung, Information Theory and Network Coding, Springer 2008.
- Robert G. Gallager, Information Theory and Reliable Communication(Paperback), Wiley, 1968.

Insegnamento**INF0011 - Teoria dell'Informazione - Parte A**

Insegnamento (inglese):	Information theory - A
CFU:	6
Settore:	INF/01 - INFORMATICA
Periodo didattico:	2
Tipologia di Attività Formativa:	B - caratterizzante
Docenti:	Marco GRANGETTO (Titolare) Matteo SERENO (Titolare)

1. Prerequisiti e Propedeuticità:**Competenze attese in ingresso**

L'organizzazione del modulo presuppone un buona conoscenza delle nozioni fornite dai corsi di probabilità e statistica e più in generale dai corsi di matematica.

2. Obiettivi formativi:

Il corso fornisce gli elementi essenziali della teoria dell'informazione che permettono di comprendere, analizzare e progettare i sistemi per la codifica e la trasmissione dell'informazione. Le competenze acquisite dagli studenti saranno la conoscenza della teoria di Shannon e delle sue molteplici applicazioni in alcuni settori dell'informatica.

3. Risultati dell'apprendimento attesi:

Al termine del corso lo studente avrà acquisito i concetti fondamentali della teoria dell'informazione e sarà in grado di applicarne i risultati alla comprensione e al progetto di sistemi di codifica e trasmissione dell'informazione

4. Modalità di verifica dell'apprendimento:

La verifica si articolerà in una prova scritta e in un esame orale. Le due parti della verifica potranno richiedere complessivamente un paio d'ore. La verifica sarà orientata ad accertare la capacità critica maturata dallo studente nella valutazione dell'applicabilità a casi di studio reali dei principi teorici studiati nelle prime tre parti costituenti gli argomenti del corso.

5. Modalità d'insegnamento:

Le lezioni in aula si svolgono in maniera tradizionale (lavagna e gesso) e, ove la natura degli argomenti esposti lo consenta, con l'ausilio del calcolatore. Le esercitazioni saranno in parte basate sulla soluzione di esercizi propedeutici alla prova d'esame.

6. Attività di supporto:

Il materiale utilizzato dai docenti durante il corso (lucidi, esercizi, articoli di approfondimento) verrà sempre reso disponibile attraverso le pagine web del corso.

7. Programma:

Il corso è strutturato in due parti.

La prima parte del corso è dedicata alla teoria dell'informazione classica: definizione dell'informazione e tipi di sorgente, concetto di entropia, la codifica di sorgente, primo teorema di Shannon (o della codifica di sorgente), codici univocamente decodificabili, ottimalità della codifica di Huffman, modelli di canale rumoroso, definizione della capacità di canale, secondo teorema di Shannon (o della codifica di canale).

La seconda parte del corso è dedicata allo studio di schemi di codifica di sorgente e canale utilizzati in molteplici applicazioni e sistemi di comunicazione. Per quanto riguarda la codifica di sorgente saranno analizzati la codifica aritmetica, la codifica di Lempel-Ziv-Welch e alcuni standard per la compressione di immagini e video. Tra gli schemi di codifica di canale verranno studiati i codici a blocco lineari, i codici ciclici e i codici convoluzionali.

8. Testi consigliati e bibliografia:

- Thomas M. Cover, Joy A. Thomas, Elements of Information Theory, Wiley, 1991 , 2006(2nd edition).
- David J. C. MacKay, Information Theory, Inference and Learning Algorithms, Cambridge University Press, 2003.
- Robert G. Gallager, Information Theory and Reliable Communication(Paperback), Wiley, 1968.

Insegnamento**INF0040 - Trattamento dei Dati e dei Processi Aziendali**

Insegnamento (inglese):	Data and Process Management in Enterprises
CFU:	6
Settore:	INF/01 - INFORMATICA
Periodo didattico:	0
Tipologia di Attività Formativa:	B - caratterizzante
Docenti:	Antonio DI LEVA (Titolare)

1. Prerequisiti e Propedeuticità:**Competenze attese in ingresso**

Lo studente deve padroneggiare i concetti fondamentali riguardanti le basi di dati (in particolare gli argomenti riguardanti i modelli concettuali e ad oggetti), e le metodologie di sviluppo dei sistemi informativi (in particolare gli argomenti riguardanti la specifica dei requisiti e il progetto dell'applicazione con UML).

2. Obiettivi formativi:

Fornire allo studente competenze generali e pratiche di base sull'analisi organizzativa e funzionale delle aziende e sulla ingegnerizzazione e re-ingegnerizzazione dei loro processi. Il corso comprende il trattamento degli aspetti sia modellistici che metodologici legati al trattamento dei dati e dei processi aziendali: • per quanto riguarda gli aspetti modellistici, verranno introdotti linguaggi e modelli per la specifica delle strutture, delle informazioni e delle funzionalità delle aziende, per modellare i processi e simularne il funzionamento al fine di valutare i parametri necessari ad una loro successiva ristrutturazione, • per quanto riguarda gli aspetti metodologici, verranno discusse le fasi principali come la raccolta e analisi dei requisiti, l'analisi dell'organizzazione e il progetto e l'ottimizzazione dei nuovi processi dell'azienda.

3. Risultati dell'apprendimento attesi:

Il corso dovrebbe fornire i concetti fondamentali e le tecniche per: • affrontare la complessità di aziende reali, • analizzarne l'organizzazione risalendo alla struttura dei processi esistenti, • acquisire competenze relative all'utilizzo di strumenti di analisi e simulazione, al fine di specificare i processi, ingegnerizzarli, e ristrutturarli.

4. Modalità di verifica dell'apprendimento:

Esame scritto sugli argomenti del corso, sviluppo di un caso di studio e discussione orale dell'elaborato. Verifiche connesse allo svolgimento assistito di applicazioni in aula dei concetti introdotti durante le lezioni frontali.

5. Modalità d'insegnamento:

Il metodo didattico consiste in lezioni frontali, integrate da esercitazioni guidate in aula. Verrà privilegiata una metodologia del tipo "insegnamento attraverso gli esempi", basando le discussioni più importanti su casi di studio concreti e su un esempio completo di sviluppo guidato. Verrà illustrato l'utilizzo di un moderno strumento per la specifica, l'animazione e la simulazione dei processi, al fine di ottenere degli indicatori di efficienza/efficacia significativi per guidare le fasi di ristrutturazione.

6. Attività di supporto:

Il corso usufruisce di un sito di supporto in e-learning che utilizza la piattaforma Moodle per la comunicazione docente-studenti, la gestione del forum per gli studenti e la distribuzione del

materiale didattico. Il sito comprende i lucidi delle lezioni, esempi di sviluppo completo di casi di studio e la distribuzione di iGrafx Process, il sistema di specifica, analisi e simulazione (ad eventi discreti) di processi aziendali.

7. Programma:

Il corso comprende la discussione di quattro parti fondamentali: 1) Introduzione ai concetti fondamentali e alle tecniche per l'analisi dei dati e dei processi aziendali. 2) Modelli di maturità dei processi. 3) Meta-modelli e linguaggi di specifica dei dati, delle informazioni, delle funzioni e dei processi delle organizzazioni. 4) Metodologie e strumenti di supporto alla rappresentazione, simulazione e reingegnerizzazione dei processi aziendali.

8. Testi consigliati e bibliografia:

Libro di testo: A. Di Leva "La gestione dell'azienda basata sui processi e i sistemi informativi aziendali" Celid, Torino (2014)

Testi consigliati: Paul Harmon "Business Process Change" 2nd ed. ISBN: 978-0-12-374152-3, ed. Elsevier (2007) David Bridgeland "Business Modeling" ISBN: 978-0-12-374151-6, ed. Elsevier (2008)

Insegnamento**MFN0899 - Valutazione delle prestazioni:
Simulazione e Modelli**

Insegnamento (inglese):	Simulation and Modelling
CFU:	9
Settore:	INF/01 - INFORMATICA
Periodo didattico:	2
Tipologia di Attività Formativa:	B - caratterizzante
Docenti:	Gianfranco BALBO (Titolare)

1. Prerequisiti e Propedeuticità:**Competenze attese in ingresso**

Rudimenti di Algebra e di Analisi. Concetti fondamentali di Probabilità e Statistica Familiarità con i contenuti dei corsi di Architetture degli Elaboratori e Sistemi Operativi

Eventuali corsi propedeutici

Architettura degli Elaboratori, Sistemi Operativi, Elementi di Probabilità e Statistica.

2. Obiettivi formativi:

Scopo di questo corso è l'introduzione alla Valutazione delle Prestazioni dei sistemi di calcolo e delle reti di telecomunicazione. Il corso si compone di due parti: la prima tratta le più importanti metodologie di analisi utilizzate per lo studio dei modelli utili per l'analisi del comportamento dei sistemi di traffico; la seconda prevede la discussione di alcuni modelli fondamentali, seguita dall'approfondimento di alcuni casi di studio presentati in letteratura.

Il livello introduttivo del corso non permette di affrontare lo studio e la modellizzazione di sistemi reali, tuttavia la preparazione fornita è sufficiente per rendere ogni studente capace di affrontare lo studio di casi reali avendo conoscenza del metodo di analisi da seguire e delle potenzialità delle tecniche disponibili. Il linguaggio e gli esempi utilizzati durante il corso si ispirano alla problematica della valutazione delle prestazioni dei sistemi di calcolo, ma le metodologie discusse hanno un ambito di applicazione ben più vasto.

I modelli discussi in questo corso sono delle rappresentazioni probabilistiche in cui vari aspetti della realtà vengono espressi sotto forma di reti di stazioni di servizio in fronte alle quali è possibile che si formino delle code a causa di fenomeni di congestione o di sincronizzazione. Lo studio del comportamento di queste reti di code viene affrontato in questo corso facendo uso di tecniche analitiche e numeriche (nei casi più semplici) e di tecniche simulative in quelli più complessi.

La parte metodologica del corso è suddivisa a sua volta in tre sezioni riguardanti l'Analisi Operazionale, la teoria delle Catene di Markov e delle code, e la Simulazione.

L'Analisi Operazionale viene discussa nella prima parte del corso per introdurre i concetti fondamentali di questa materia e per affrontare la modellazione e la soluzione di casi relativamente semplici. Quando le caratteristiche probabilistiche dei modelli da analizzare si complicano, il corso introduce le Catene di Markov ed illustra come questo modello matematico possa permettere di affrontare con eleganza problemi intrinsecamente molto complessi, ma a spesa di un costo computazionale generalmente molto elevato. Quando anche le Catene di Markov risultano inadeguate per affrontare le problematiche dei sistemi reali si fa ricorso alla Simulazione ad Eventi Discreti.

La simulazione del comportamento di un modello probabilistico consiste nella scrittura di un

programma capace di riprodurre (con un certo livello di astrazione) le modalità di funzionamento del modello stesso. L'esecuzione di questo programma corrisponde ad una evoluzione del modello a partire da un certo stato iniziale per giungere ad un determinato stato finale. Questa esecuzione è guidata da generatori di numeri casuali e corrisponde quindi ad una delle possibili evoluzioni del modello. Le misure eseguite durante la simulazione diventano pertanto delle istanze di variabili casuali e sono oggetto di analisi statistica per fornire stime intervallari (intervalli di confidenza) degli indici di prestazione del modello stesso. Sviluppate le conoscenze delle tecniche di base utilizzate per l'analisi di questi modelli, il corso sposta l'attenzione su alcune applicazioni di riferimento prese dal contesto delle reti di telecomunicazione e dei sistemi di calcolo distribuiti per affrontare la modellazione di alcune loro componenti e per valutarne le prestazioni. Questi casi di studio, presi dalla letteratura, sono discussi con lo scopo di fornire allo studente un'idea dell'uso reale di queste metodologie e della complessità della loro applicazione in casi concreti.

Parte integrante del corso sono una serie di esercizi che vengono periodicamente assegnati ed un progetto finale che viene discusso in sede di esame.

3. Risultati dell'apprendimento attesi:

Conoscenza delle tecniche fondamentali per lo studio delle prestazioni di sistemi di calcolo. Sviluppo di competenze approfondite per un uso intelligente ed informato di software specializzati comunemente impiegati in industria

4. Modalità di verifica dell'apprendimento:

L'esame orale comporta la discussione dei risultati ottenuti durante il progetto (vedi dopo) e la risposta ad alcune domande di carattere più teorico.

PROGETTO I contenuti del corso vengono sintetizzati nella esecuzione di un progetto finale che consiste nello studio di un modello "quasi reale" utilizzando tutte le tecniche di analisi apprese durante il corso. Il progetto si configura come un esercizio "libero" piuttosto consistente, in cui devono essere posti in risalto anche i pregi ed i difetti delle tecniche usate.

L'esecuzione del progetto è condizione necessaria per potere sostenere l'esame del corso. Il progetto deve essere completato compilando una relazione in cui vengono descritti il problema affrontato, le ipotesi introdotte, il metodo di analisi utilizzato e la discussione dei risultati. Il testo del progetto potrà fare riferimento ai casi di studio analizzati durante la parte finale del corso.

5. Modalità d'insegnamento:

La presentazione dei contenuti del corso avviene utilizzando prevalentemente la lavagna. Le lezioni del corso si alternano alla discussione della soluzione di esercizi che vengono assegnati quasi settimanalmente. Alla fine del corso viene assegnato un progetto che deve essere portato all'esame orale per essere utilizzato come base per la conduzione dello stesso. L'esecuzione puntuale degli esercizi assegnati è consigliabile per giungere alla fine del corso in condizione di preparazione tali da permettere una veloce preparazione del progetto finale e quindi il superamento dell'esame orale nella prima sessione successiva di esami.

6. Attività di supporto:

La materia trattata nel corso è discussa in maniera ampia ed approfondita all'interno di una serie di "dispense" che il docente mette a disposizione degli studenti in concomitanza con la presentazione dei vari argomenti.

Le dispense stanno assumendo la forma di un libro che sarà reso disponibile on-line agli studenti del corso.

Materiale aggiuntivo (sotto forma di copie elettroniche di alcune trasparenze usate dal docente per discutere alcuni aspetti specifici del corso) è reso disponibile durante il prosieguo del corso.

7. Programma:

PARTE I: Metodologia

- Introduzione o Modellistica dei sistemi di calcolo ed indici di prestazione o Formalismi per la

definizione dei modelli

- Analisi Operazionale o Introduzione, entità misurabili e variabili operazionali o Analisi del flusso nelle reti di code o Equazioni di bilanciamento o Reti di code con soluzioni sotto forma di prodotto o Algoritmi per il calcolo della soluzione sotto forma di prodotto
- Catene di Markov o Richiami di calcolo delle probabilità o Processi Stocastici o Proprietà Markoviana o Classificazione degli stati o Equazioni di Bilanciamento o Soluzioni a regime ed in transitorio o Processi di nascita e morte o Principi di teoria delle code classica o Soluzione di alcuni sistemi a coda elementari
- Simulazione o Introduzione alla simulazione ad eventi discreti o Costruzione di un programma simulatore o Cenni di statistica elementare o Numeri casuali o Test statistici o Metodi per la generazione di istanze di variabili casuali o Strutture dati ed organizzazione di un programma simulatore o Analisi statistica dei risultati e convalida dei modelli di simulazione.

PARTE II: Modelli

- Formalismi per la costruzione ed analisi di modelli ad eventi discreti • Central Server Network; • Erlang loss model; • Casi di Studio tratti dalla letteratura.

8. Testi consigliati e bibliografia:

Leonard Kleinrock, "Sistemi a Coda: Introduzione alla Teoria delle Code", Hoepli, 1992.

Hisashi Kobayashi, "Modeling and Analysis: An Introduction to System Performance Evaluation Methodology", Addison-Wesley, 1978.

Giuseppe Iazeolla, "Introduzione alla Simulazione Discreta", Boringhieri, 1980.

Sheldon M. Ross, "A Course in Simulation", Macmillan, 1990.

George S. Fishman, "Principles of Discrete Event Digital Simulation", John Wiley & Sons, 1978.

Insegnamento**MFN1361 - Valutazione delle prestazioni:
Simulazione e Modelli - Parte A**

Insegnamento (inglese):	Simulation and Modelling - A
CFU:	6
Settore:	INF/01 - INFORMATICA
Periodo didattico:	2
Tipologia di Attività Formativa:	B - caratterizzante
Docenti:	Gianfranco BALBO (Titolare)

1. Prerequisiti e Propedeuticità:**Competenze attese in ingresso**

Rudimenti di Algebra e di Analisi. Concetti fondamentali di Probabilità e Statistica Familiarità con i contenuti dei corsi di Architetture degli Elaboratori e Sistemi Operativi

Eventuali corsi propedeutici

Architettura degli Elaboratori, Sistemi Operativi, Elementi di Probabilità e Statistica.

2. Obiettivi formativi:

Scopo di questo corso è l'introduzione alla Valutazione delle Prestazioni dei sistemi di calcolo e delle reti di telecomunicazione. Il corso tratta le più importanti metodologie di analisi utilizzate per lo studio dei modelli utili per l'analisi del comportamento dei sistemi di traffico.

Il livello introduttivo del corso non permette di affrontare lo studio e la modellizzazione di sistemi reali, tuttavia la preparazione fornita è sufficiente per rendere ogni studente capace di affrontare lo studio di casi reali avendo conoscenza del metodo di analisi da seguire e delle potenzialità delle tecniche disponibili. Il linguaggio e gli esempi utilizzati durante il corso si ispirano alla problematica della valutazione delle prestazioni dei sistemi di calcolo, ma le metodologie discusse hanno un ambito di applicazione ben più vasto.

I modelli discussi in questo corso sono delle rappresentazioni probabilistiche in cui vari aspetti della realtà vengono espressi sotto forma di reti di stazioni di servizio in fronte alle quali è possibile che si formino delle code a causa di fenomeni di congestione o di sincronizzazione. Lo studio del comportamento di queste reti di code viene affrontato in questo corso facendo uso di tecniche analitiche e numeriche (nei casi più semplici) e di tecniche simulative in quelli più complessi.

L'Analisi Operazionale viene discussa all'inizio del corso per introdurre i concetti fondamentali di questa materia e per affrontare la modellazione e la soluzione di casi relativamente semplici. Quando le caratteristiche probabilistiche dei modelli da analizzare si complicano, il corso introduce le Catene di Markov ed illustra come questo modello matematico possa permettere di affrontare con eleganza problemi intrinsecamente molto complessi derivanti dalla Teoria delle Code, ma a spesa di un costo computazionale generalmente molto elevato. Quando anche le Catene di Markov risultano inadeguate per affrontare le problematiche dei sistemi reali si fa ricorso alla Simulazione.

La simulazione del comportamento di un modello probabilistico consiste nella scrittura di un programma che riproduca (con un certo livello di astrazione) le modalità di funzionamento del modello stesso. L'esecuzione di questo programma corrisponde ad una evoluzione del modello a partire da un certo stato iniziale per giungere ad un determinato stato finale. Questa esecuzione è guidata da generatori di numeri casuali e corrisponde quindi ad una delle possibili evoluzioni del modello. Le misure eseguite durante la simulazione sono oggetto di analisi statistica per fornire

una stima degli indici di prestazione del modello stesso.

Parte integrante del corso sono una serie di esercizi che vengono periodicamente assegnati ed un progetto finale che viene discusso in sede di esame.

3. Risultati dell'apprendimento attesi:

Conoscenza delle tecniche fondamentali per lo studio delle prestazioni di sistemi di calcolo. Sviluppo di competenze approfondite per un uso intelligente ed informato di software specializzati comunemente impiegati in industria

4. Modalità di verifica dell'apprendimento:

L'esame orale comporta la discussione dei risultati ottenuti durante il progetto (vedi dopo) e la risposta ad alcune domande di carattere più teorico.

PROGETTO I contenuti del corso vengono sintetizzati nella esecuzione di un progetto finale che consiste nello studio di un modello "quasi reale" utilizzando tutte le tecniche di analisi apprese durante il corso. Il progetto si configura come un esercizio "libero" piuttosto consistente, in cui devono essere posti in risalto anche i pregi ed i difetti delle tecniche usate.

L'esecuzione del progetto è condizione necessaria per potere sostenere l'esame del corso. Il progetto deve essere completato compilando una relazione in cui vengono descritti il problema affrontato, le ipotesi introdotte, il metodo di analisi utilizzato e la discussione dei risultati. Il testo del progetto potrà fare riferimento ai casi di studio analizzati durante la parte finale del corso.

5. Modalità d'insegnamento:

La presentazione dei contenuti del corso avviene utilizzando prevalentemente la lavagna. Le lezioni del corso si alternano alla discussione della soluzione di esercizi che vengono assegnati quasi settimanalmente. Alla fine del corso viene assegnato un progetto che deve essere portato all'esame orale per essere utilizzato come base per la conduzione dello stesso. L'esecuzione puntuale degli esercizi assegnati è consigliabile per giungere alla fine del corso in condizione di preparazione tali da permettere una veloce preparazione del progetto finale e quindi il superamento dell'esame orale nella prima sessione successiva di esami.

6. Attività di supporto:

La materia trattata nel corso è discussa in maniera ampia ed approfondita all'interno di una serie di "dispense" che il docente mette a disposizione degli studenti in concomitanza con la presentazione dei vari argomenti.

Le dispense stanno assumendo la forma di un libro che sarà reso disponibile on-line agli studenti del corso.

Materiale aggiuntivo (sotto forma di copie elettroniche di alcune trasparenze usate dal docente per discutere alcuni aspetti specifici del corso) è reso disponibile durante il prosieguo del corso.

7. Programma:

Metodologia

- Introduzione o Modellistica dei sistemi di calcolo ed indici di prestazione o Formalismi per la definizione dei modelli
- Analisi Operazionale o Introduzione, entità misurabili e variabili operazionali o Analisi del flusso nelle reti di code o Equazioni di bilanciamento o Reti di code con soluzioni sotto forma di prodotto o Algoritmi per il calcolo della soluzione sotto forma di prodotto
- Catene di Markov o Richiami di calcolo delle probabilità o Processi Stocastici o Proprietà Markoviana o Classificazione degli stati o Equazioni di Bilanciamento o Soluzioni a regime ed in transitorio o Processi di nascita e morte o Principi di teoria delle code classica o Soluzione di alcuni sistemi a coda elementari

• Simulazione o Introduzione alla simulazione ad eventi discreti o Costruzione di un programma simulatore o Cenni di statistica elementare o Numeri casuali o Test statistici o Metodi per la generazione di istanze di variabili casuali o Strutture dati ed organizzazione di un programma simulatore o Analisi statistica dei risultati e convalida dei modelli di simulazione.

8. Testi consigliati e bibliografia:

Leonard Kleinrock, "Sistemi a Coda: Introduzione alla Teoria delle Code", Hoepli, 1992.

Hisashi Kobayashi, "Modeling and Analysis: An Introduction to System Performance Evaluation Methodology", Addison-Wesley, 1978.

Giuseppe Iazeolla, "Introduzione alla Simulazione Discreta", Boringhieri, 1980.

Sheldon M. Ross, "A Course in Simulation", Macmillan, 1990.

Insegnamento**MFN0959 - Verifica dei Programmi Concorrenti**

Insegnamento (inglese):	Software Reliability Methods
CFU:	9
Settore:	INF/01 - INFORMATICA
Periodo didattico:	2
Tipologia di Attività Formativa:	B - caratterizzante
Docenti:	Susanna DONATELLI (Titolare)

1. Prerequisiti e Propedeuticità:**Competenze attese in ingresso**

Competenze e conoscenze della laurea L31, con particolare attenzione alla capacità di astrazione e alla conoscenza dei principi di base della concorrenza.

Eventuali corsi propedeutici

Corso di base su linguaggi formali e sistemi operativi (parte sui processi e sulla comunicazione). IL corso di MCAD (modelli concorrenti e algoritmi distribuiti) non è propedeutico, ma lo studente/studentessa che segua entrambi i corsi potrà sicuramente avere una migliore visione di insieme degli argomenti dei due corsi.

2. Obiettivi formativi:

Obiettivo del corso è di fornire gli strumenti, teorici e pratici, necessari alla verifica dei sistemi, ed in particolare del software. Per raggiungere tale obiettivo studieremo alcuni paradigmi di base per la specifica di processi distribuiti, focalizzando l'attenzione sulle capacità modellistiche e sugli strumenti di verifica di proprietà di buon comportamento. Studieremo inoltre la verifica dei sistemi in cui il corretto funzionamento dipende da caratteristiche di tempo.

In particolare il corso ha come obiettivo di far acquisire le competenze teoriche legate ai modelli della concorrenza e di farle applicare per risolvere problemi concreti di verifica del software. Le modalità di insegnamento favoriranno l'acquisizione, da parte degli studenti, di capacità autonoma di apprendimento, (comprensione di articoli scientifici sugli argomenti del corso), di abilità comunicative (relazioni associate agli esercizi di laboratorio e discussioni in classe) e di autonomia di giudizio (grazie a laboratori in cui gli studenti debbono organizzare esperimenti e valutazione degli stessi)

3. Risultati dell'apprendimento attesi:

Alla fine del corso lo studente sarà in grado di specificare sistemi concorrenti usando linguaggi formali e di utilizzare strumenti software per la verifica di proprietà del sistema tramite verifica di proprietà del modello. In particolare lo studente acquisirà familiarità con i seguenti strumenti: GreatSPN, NuSMV e Uppaal. Oltre alle classiche proprietà dei sistemi distribuiti quali assenza di deadlock, fairness e liveness, lo studente sarà in grado di definire e verificare proprietà in logica temporale quali ad esempio: "se il processo P manda un messaggio, allora non invierà il prossimo messaggio sino a che non riceve un acknowledgment", oppure "se il processo P manda un messaggio, riceverà un acknowledge entro 5 unità di tempo"

4. Modalità di verifica dell'apprendimento:

Il corso ha due modalità di esame, a scelta dello studente/studentessa, a seconda che egli/ella voglia approfondire maggiormente gli obiettivi formativi legati agli aspetti di definizione e confronto dei paradigmi di modellazione concorrente o gli aspetti legati agli strumenti di verifica. Tipo 1: esame orale su argomenti di base del corso e un set esteso di esercizi di laboratorio da discutere con il docente Tipo 2: esame orale su tutti gli argomenti del corso e un insieme ridotto di

esercizi di laboratorio da discutere con il docente Per l'esame di tipo 1 la valutazione è basata per l'80% sul laboratorio, ma è comunque richiesta la sufficienza anche nella parte di esame orale per poter superare l'esame. Per l'esame di tipo 2 la valutazione è basata per l'80% sull'orale, ma è comunque richiesta la sufficienza anche nella parte di laboratorio per poter superare l'esame. L'individuazione degli argomenti di base per l'esame di tipo 1 avviene durante il corso, alla fine di ogni macro argomento e viene pubblicato sul sito moodle del corso. Gli esercizi di laboratorio vengono assegnati durante il corso (nella forma "lunga" per l'esame di tipo 1, e nella forma "corta" per l'esame di tipo 2). Gli esercizi sono resi disponibili sul sito moodle del corso.

5. Modalità d'insegnamento:

Lezioni frontali, con utilizzo di strumenti di verifica da parte del docente. Durante il corso vengono assegnati esercizi e mini progetti che gli studenti sono invitati a svolgere durante il corso, a scadenze concordate. Almeno 30 ore vengono dedicate a lezioni/laboratorio in cui vengono presentati gli strumenti di verifica e vengono discussi i problemi che gli studenti hanno incontrato nello svolgimento dei mini progetti. In queste lezioni gli studenti sono invitati a presentare le loro soluzioni che vengono discusse dal docente e dagli altri studenti

6. Attività di supporto:

Il corso utilizza il sito Moodle (voce e-learning sul sito web del corso di studi) per la distribuzione del materiale e degli esercizi (e la relativa consegna) e per la messa a disposizione di un forum di discussione fra studenti

7. Programma:

- Introduzione alla verifica dei sistemi
- Un primo linguaggio di specifica, le Reti di Petri (reti posto transizione e reti colorate): definizione, proprietà e tool disponibili (GreatSPN).
- Un secondo linguaggio di specifica, le algebre dei processi: definizione, proprietà di CCS e CSP
- Esprimere proprietà: le logiche temporali lineari (LTL) e branching (CTL) e i relativi tool (NuSMV e RGMEDD)
- Esprimere il tempo: gli automi temporizzati, le logiche branching temporizzate e relativi tool (Uppaal)

8. Testi consigliati e bibliografia:

1. Doron Peled, Software Reliability Models, Springer and Verlag, ISBN: 978-1-4419-2876-4 (Print) 978-1-4757-3540-6 (Online)

2. Christel Baier and Jost-Pieter Katoen, Model Checking – MIT Press

Insegnamento**MFN1360 - Verifica dei Programmi Concorrenti - Parte A**

Insegnamento (inglese):	Software Reliability Methods - A
CFU:	6
Settore:	INF/01 - INFORMATICA
Periodo didattico:	2
Tipologia di Attività Formativa:	B - caratterizzante
Docenti:	Susanna DONATELLI (Titolare)

1. Prerequisiti e Propedeuticità:**Competenze attese in ingresso**

Competenze e conoscenze della laurea L31, con particolare attenzione alla capacità di astrazione e alla conoscenza dei principi di base della concorrenza.

Eventuali corsi propedeutici

Corso di base su linguaggi formali e sistemi operativi (parte sui processi e sulla comunicazione). Il corso di MCAD (modelli concorrenti e algoritmi distribuiti) non è propedeutico, ma lo studente/studentessa che segua entrambi i corsi potrà sicuramente avere una migliore visione di insieme degli argomenti dei due corsi.

2. Obiettivi formativi:

Obiettivo del corso è di fornire gli strumenti, teorici e pratici, necessari alla verifica dei sistemi, ed in particolare del software. Per raggiungere tale obiettivo studieremo alcuni paradigmi di base per la specifica di processi distribuiti, focalizzando l'attenzione sulle capacità modellistiche e sugli strumenti di verifica di proprietà di buon comportamento. In particolare il corso ha come obiettivo di far acquisire le competenze teoriche legate ai modelli della concorrenza e di farle applicare per risolvere problemi concreti di verifica del software. Le modalità di insegnamento favoriranno l'acquisizione, da parte degli studenti, di capacità autonoma di apprendimento, (comprensione di articoli scientifici sugli argomenti del corso), di abilità comunicative (relazioni associate agli esercizi di laboratorio e discussioni in classe) e di autonomia di giudizio (grazie a laboratori in cui gli studenti debbono organizzare esperimenti e valutazione degli stessi)

3. Risultati dell'apprendimento attesi:

Alla fine del corso lo studente sarà in grado di specificare sistemi concorrenti usando linguaggi formali e di utilizzare strumenti software per la verifica di proprietà del sistema tramite verifica di proprietà del modello. In particolare lo studente acquisirà familiarità con i seguenti strumenti: GreatSPN, NuSMV e Uppaal. Oltre alle classiche proprietà dei sistemi distribuiti quali assenza di deadlock, fairness e liveness, lo studente sarà in grado di definire e verificare proprietà in logica temporale quali ad esempio: "se il processo P manda un messaggio, allora non invierà il prossimo messaggio sino a che non riceve un acknowledgment"

4. Modalità di verifica dell'apprendimento:

Il corso ha due modalità di esame, a scelta dello studente/studentessa, a seconda che egli/ella voglia approfondire maggiormente gli obiettivi formativi legati agli aspetti di definizione e confronto dei paradigmi di modellazione concorrente o gli aspetti legati agli strumenti di verifica. Tipo 1: esame orale su argomenti di base del corso e un set esteso di esercizi di laboratorio da discutere con il docente Tipo 2: esame orale su tutti gli argomenti del corso e un insieme ridotto di esercizi di laboratorio da discutere con il docente Per l'esame di tipo 1 la valutazione è basata per l'80% sul laboratorio, ma è comunque richiesta la sufficienza anche nella parte di esame orale per poter superare l'esame. Per l'esame di tipo 2 la valutazione è basata per l'80% sull'orale, ma è

comunque richiesta la sufficienza anche nella parte di laboratorio per poter superare l'esame. L'individuazione degli argomenti di base per l'esame di tipo 1 avviene durante il corso, alla fine di ogni macro argomento e viene pubblicato sul sito moodle del corso. Gli esercizi di laboratorio vengono assegnati durante il corso (nella forma "lunga" per l'esame di tipo 1, e nella forma "corta" per l'esame di tipo 2). Gli esercizi sono resi disponibili sul sito moodle del corso.

5. Modalità d'insegnamento:

Lezioni frontali, con utilizzo di strumenti di verifica da parte del docente. Durante il corso vengono assegnati esercizi e mini progetti che gli studenti sono invitati a svolgere durante il corso, a scadenze concordate. Almeno 20 ore vengono dedicate a lezioni/laboratorio in cui vengono presentati gli strumenti di verifica e vengono discussi i problemi che gli studenti hanno incontrato nello svolgimento dei mini progetti. In queste lezioni gli studenti sono invitati a presentare le loro soluzioni che vengono discusse dal docente e dagli altri studenti

6. Attività di supporto:

Il corso utilizza il sito Moodle (voce e-learning sul sito web del corso di studi) per la distribuzione del materiale e degli esercizi (e la relativa consegna) e per la messa a disposizione di un forum di discussione fra studenti. Non c'è un sito separato per la parte A del corso, gli studenti fanno riferimento al sito moodle del corso da 9 cfu

7. Programma:

- Introduzione alla verifica dei sistemi
- Un primo linguaggio di specifica, le Reti di Petri (reti posto transizione): definizione, proprietà e tool disponibili (GreatSPN).
- Un secondo linguaggio di specifica, le algebre dei processi: definizione, proprietà di CCS e CSP
- Esprimere proprietà: le logiche temporali lineari (LTL) e branching (CTL) e i relativi tool (NuSMV e RGMEDD)

NOTA: rispetto al corso da 9 cfu gli argomenti mancanti sono le reti di Petri colorate (e relativi esercizi) e la parte sul tempo (timed automata e Uppaal

8. Testi consigliati e bibliografia:

1. Doron Peled, Software Reliability Models, Springer and Verlag, ISBN: 978-1-4419-2876-4 (Print) 978-1-4757-3540-6 (Online)

2.Christel Baier and Jost-Pieter Katoen, Model Checking – MIT Press