

**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 2014/2015



di.unito.it
DIPARTIMENTO DI INFORMATICA



UNIVERSITÀ
DEGLI STUDI
DI TORINO



(Bollino 2014 in fase di erogazione)

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 2014”. La seconda parte descrive invece gli insegnamenti offerti per il 2014/2015 (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 2014), che trovano nel Manifesto la descrizione del loro percorso biennale, mentre la seconda parte è di interesse anche per gli studenti della coorte precedente (2013) 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 immatricolazione e disponibile su questo stesso sito.

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

1. La durata della Laurea Magistrale è di due anni accademici. Il corso di studi è articolato in tre diversi indirizzi, 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.
2. L’ammissione al corso di Laurea Magistrale richiede (almeno) un titolo di laurea triennale.
3. L’ammissione al corso di Laurea Magistrale in Informatica viene deliberata da apposita commissione dopo la valutazione dell’adeguatezza della carriera (tipo di studi triennali effettuati ed eventuali altre informazioni pertinenti) ed un colloquio con il candidato. La carriera è sempre adeguata per i laureati triennali in Informatica.
4. La Laurea magistrale dà accesso, previo esame di selezione, al Dottorato di Ricerca in Informatica, della durata di tre anni.
5. La Laurea Magistrale dà anche 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.
6. Dal 2012/2013 non è più previsto il “libretto degli esami” in forma cartacea, ma solo in forma elettronica. Analogamente l’iscrizione e la registrazione del voto degli esami avverrà tramite una procedura telematica, attiva sul sito web dell’Università di Torino.
7. Tutte le informazioni non riportate in questa guida (come i programmi dettagliati dei corsi, gli orari, la localizzazione delle aule, tasse e scadenze) sono reperibili dal sito Internet del Corso di Studi, all’indirizzo www.informatica.unito.it, o sul sito di Ateneo www.unito.it
8. Una diversa descrizione del Corso di Studi si può trovare sul sito del Ministero, [link a sito MIUR \(diretto al nostro corso di studi\)](#), oppure [link a sito MIUR \(tutti i corsi di studio\)](#).

Buona lettura!

Susanna Donatelli, PhD,
Professore Ordinario di Informatica,
Presidente del Corso di Studi in Informatica
(Tel.0116706711 e-mail presccs@educ.di.unito.it)

INDICE

Prefazione	2
Prima parte: Manifesto degli studi	4
Obiettivi e sbocchi professionali	4
Obiettivi del Corso di Laurea	4
Sbocchi professionali	4
Certificazione di Qualità	5
Organizzazione della didattica	5
Introduzione	5
Requisiti e modalità di ammissione	6
Periodi di lezione e esami	7
Piano Carriera	8
Struttura del corso di studio e curricula previsti	8
Indicazioni comuni a tutti i curricula	8
Struttura dell'offerta formativa rispetto al RAD della classe	9
Curriculum "Realtà Virtuale e Multimedialità"	11
Curriculum "Reti e Sistemi informatici"	13
Curriculum "Sistemi per il Trattamento dell'Informazione"	15
Dopo la magistrale: il dottorato di ricerca	18
Il Dottorato di Ricerca in Informatica presso il Dipartimento di Informatica	18
Organizzazione del Dottorato	18
Informazioni	19
Supporto agli studi	19
Erasmus	19
Biblioteca	19
Supporto on-line ai corsi	20
Aule e laboratori	20
Zone studio	20
Supporto agli studenti Disabili	21
Supporto agli studenti con DSA (Disturbi Specifici di Apprendimento)	22
Indirizzi utili	23
Seconda Parte: Programmi e docenti dei Corsi per l'A.A. 14/15	24
Programmi, docenti e semestri per i corsi attivi nel 14/15.	24
Corsi mutuati attivi nell'A.A. 14/15:	26
Programmi e docenti previsti per i corsi ad anni alterni non tenuti nel 14/15 o di futura attivazione:	27
Codici per corsi esterni all'Ateneo.	29

Prima parte: Manifesto degli studi

Obiettivi e sbocchi professionali

Obiettivi del Corso di Laurea

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 studi 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 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 studi, 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, logica della computazione, realtà virtuale e multimedialità, reti e sistemi informatici, sistemi per il trattamento dell'informazione. A partire da questi temi sono stati al momento attivati tre indirizzi: 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 e di supporto alle reti di comunicazione, 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 sistemi informatici complessi per gestire i propri dati e processi. Le competenze acquisite durante il corso di studi 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**: analisti e progettisti di software, analisti di sistema, specialisti in reti e comunicazioni informatiche, 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 in Informatica italiani. Il nostro Corso di Laurea ha già ricevuto tale certificato nazionale di qualità negli anni scorsi ed è in corso la procedura di acquisizione per il 2014-2015.

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

Inoltre, a partire dal 2005, il Corso di Laurea è stato accreditato dalla Regione Piemonte, a seguito di una visita e di una verifica sulla buona organizzazione e sul buon funzionamento del Corso

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 triennale 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 Studi 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

La normativa vigente richiede che prima dell'iscrizione venga sostenuto un colloquio volto a stabilire tanto l'adeguatezza del curriculum che la personale preparazione del candidato.

Per quanto riguarda l'adeguatezza del curriculum questo è garantita 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 e automatica, oltre che nel settore matematico.

Qualora l'adeguatezza del Curriculum sia riconosciuta, il candidato sosterrà il colloquio volto a valutare la personale preparazione e definire il piano di studio.

Le materie oggetto del colloquio sono le seguenti:

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

I colloqui si svolgeranno periodicamente, in aule aperte al pubblico [previa comunicazione](#), sul sito del Corso di Studi, di fronte alla Commissione per l'ammissione; non sarà consentito sostenere il colloquio di ammissione più di due volte per ciascun anno accademico.

L'ammissione avviene secondo le modalità descritte sul sito web del corso di laurea, con calendari disponibili alla stessa pagina, aggiornati in itinere a seconda del numero di richieste da evadere. L'ammissione segue lo [schema di ammissione](#) definito dall'Ateneo per tutti i corsi di laurea magistrale ad accesso libero.

L'iter prevede che gli studenti debbano presentare domanda di verifica dei requisiti curriculari (domanda di ammissione preliminare) all'apposita commissione del corso di studi. Le strutture didattiche comunicano gli esiti delle domande di ammissione preliminare agli studenti e fissano i colloqui per la verifica della preparazione personale. Gli studenti per i quali l'esito della domanda di ammissione preliminare sia **positivo** devono compilare la domanda di iscrizione on line dal 1° settembre 2014 al 23 dicembre 2014 e consegnarla, debitamente firmata e corredata dei documenti richiesti, dal 1° settembre 2014 e **entro il 23 dicembre 2014**, presso la Segreteria Studenti di riferimento (si veda: <http://di.unito.it/segreteriaStudenti>). Al fine di consentire l'iscrizione ai corsi di laurea magistrale entro i termini, gli studenti possono presentare domanda di ammissione preliminare anche prima di aver conseguito il titolo triennale. Tale pratica è particolarmente consigliata per gli studenti che si laureeranno nell'autunno 2014.

Per gli studenti che conseguano il titolo *oltre la scadenza dell'iscrizione alla magistrale*, il Senato Accademico ha deliberato in merito alla possibilità di iscriversi a corsi singoli, sino ad un massimo di 30 cfu, e di far valere poi i cfu così acquisiti come parte della carriera di una laurea magistrale:

omissis...

ISCRIZIONE A CORSI SINGOLI

m) l'iscrizione a corsi singoli/singoli insegnamenti per i corsi di studio ad accesso libero avviene contestualmente al pagamento della tassa prevista e può essere effettuata per un massimo di 30 cfu per anno accademico.

omissis....

Periodi di lezione e esami

L'Anno Accademico del Corso di Studi 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 2014/2015 le date di inizio e di fine del periodo di lezioni sono le seguenti:

Primo semestre: 29 settembre 2014 – 16 gennaio 2015

Sessione invernale di esami: 19 gennaio 2015 – 20 febbraio 2015

Secondo semestre: 23 febbraio 2015 - 05 giugno 2015

Sessione estiva di esami: 08 giugno 2015 – 31 luglio 2015

Sessione autunnale di esami: 01 settembre 2015 – inizio lezioni a.a. 2015-2016

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 i corsi con Laboratorio.

Per ogni unità didattica oggetto d'esame, sono previsti cinque appelli d'esame all'anno. Le date e gli appelli degli esami saranno 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

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 novembre 2014 al **31 gennaio 2015**.

Lo studente deve presentare il proprio piano carriera nei tempi descritti nella sezione “date e scadenze”. Lo studente dovrà selezionare tutti i corsi del piano carriera così come stabiliti al momento dell'ammissione. Ulteriori variazioni andranno **obbligatoriamente** concordate **preventivamente** con la commissione piani di studio. Attenzione, 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 prima e con la commissione piani di studio, eventualmente, poi. Pertanto eventuali deviazioni dal piano concordato verranno rilevate solo al momento della vostra iscrizione all'esame di laurea: 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 studi.

Struttura del corso di studio e curricula previsti

Nel seguito viene descritta la struttura del corso di studi, comune ai tre curricula, e come tale struttura sia declinata negli insegnamenti per i tre curricula attivati per l'A.A. 2014/2015. Vengono come prima cosa delineati alcuni aspetti comuni ai tre curricula, 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 può inserire sino a 21 crediti liberi, arrivando ad un totale di 129 cfu per l'intero percorso formativo.

2. Lo studente non può inserire nel Piano di Studi insegnamenti che hanno argomenti significativamente sovrapposti ad argomenti di corsi già acquisiti nel percorso triennale.

3. Durante il colloquio verrà verificato nel dettaglio il percorso di I Livello e la sua adeguatezza, precedentemente valutata dalla Commissione: 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. I corsi etichettati come "... – Parte A" si riferiscono a moduli da 6 cfu che sono mutuati dai rispettivi corsi 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 indirizzi è indicato l'anno di attivazione. Gli esami vengono assicurati anche per gli anni in cui il corso non si tiene.

6. I programmi dei corsi e i semestri di svolgimento li potete trovare nella [seconda parte della guida](#), quella relativa all'offerta per il 14/15. Per completezza sono elencati, separatamente, anche i corsi 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, definita prevede una strutturazione con 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 studi della classe LM-18, qualunque sia la sede. Il corso di studi magistrali 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, 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 gli indirizzi del corso di studi, 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 indirizzo, 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 del corso 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 2014

Codice	Insegnamento	SSD	CFU	Anno	Anno consigliato
CURRICULUM Realtà Virtuale e Multimedialità -- percorso 102					
Formazione caratterizzante:					
mfn0972	Elaborazione di Immagini e Visione Artificiale	inf/01	9	1	1
mfn0973	Modellazione Grafica	inf/01	9	1	1
mfn0978	Sistemi di Realtà Virtuale	inf/01	9	2	2
mfn0961	Teoria dell'Informazione	inf/01	9	1	1
Tre insegnamenti a scelta tra:					
inf0007	Analisi e Visualizzazione di Reti Complesse	inf/01	9	2	2
mfn0943	Apprendimento Automatico e Analisi Intelligente dei Dati ANNI ALTERNI 14-15 SI	inf/01	9	1	1
mfn0947	Basi di Dati Multimediali 14-15 TACE	inf/01	9	2	2
mfn0942	Intelligenza Artificiale e Laboratorio	inf/01	9	1 o 2	1 o 2
mfn0940	Modelli Avanzati e Architetture di Basi di Dati	inf/01	9	1	1
mfn0949	Sistemi Cognitivi	inf/01	9	1 o 2	1 o 2
mfn0959	Verifica dei Programmi Concorrenti	inf/01	9	1 o 2	1 o 2
Un insegnamento a scelta tra:					
mfn0974	Elaborazione Digitale Audio e Musica	inf/01	6	1	1
mfn0960	Modelli Concorrenti e Algoritmi distribuiti	inf/01	6	1	1
inf0008	Programmazione per Dispositivi Mobili	inf/01	6	2	2
mfn0977	Reti Neurali	inf/01	6	2	2
mfn0795	Sistemi di Calcolo Paralleli e Distribuiti	inf/01	6	1 o 2	1 o 2
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	inf/01	6	1	1
mfn0987	Istituzioni di Sistemi Intelligenti	inf/01	6	1	1
mfn0989	Istituzioni di Sviluppo Software	inf/01	6	1	1
mfn0990	Istituzioni di Tecnologie Web	inf/01	6	1	1
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)					
mfn0962	Metodi Numerici	mat/08	6	1	1
mfn0970	Fisica per Applicazioni di Realtà Virtuale	fis/01	6	1	1

mfn0971	Complementi di Analisi e Probabilità	3 mat/05, 3 mat/06	6	1	1
mfn1001	Istituzioni di Economia e Gestione dell'Impresa	secs-p/08	6	1	1
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	L- Lin/12	3	1 o 2	1 o 2
mfn0979	Prova Finale		24	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 i corsi 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

I corsi si dividono in:

- corsi obbligatori (per 18 cfu) e corsi opzionali in ristretto gruppo di scelta (per 33 cfu)
- corsi opzionali in un vasto gruppo di scelta (18cfu) e corsi liberi (12 cfu)
- corsi 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 2014

Codice	Insegnamento	SSD	CFU	anno	Anno consigliato
CURRICULUM Reti e Sistemi Informatici - percorso 103					
Blocco 1	Formazione caratterizzante:				
inf0006	Gestione delle Reti 14-15 NON ATTIVO	inf/01	6	2	2
inf0009	Reti II	inf/01	6	1 o 2	1
mfn0952	Sicurezza II	inf/01	6	1	1
Blocco 2	Un insegnamento a scelta tra:				
mfn0899	Valutazione delle prestazioni: Simulazione e Modelli	inf/01	9	1	1
mfn0959	Verifica dei Programmi Concorrenti	inf/01	9	1 o 2	1
Blocco 3	Due insegnamenti a scelta tra:				
inf0007	Analisi e Visualizzazione di Reti Complesse	inf/01	9	2	2
mfn0947	Basi di Dati Multimediali 14-15 NON ATTIVO	inf/01	9	2	2
mfn0941	Sviluppo Software per Componenti e Servizi Web	inf/01	9	1 o 2	2
mfn0961	Teoria dell'Informazione	inf/01	9	1	1
Blocco 4	Un insegnamento a scelta tra:				
mfn0969	Architettura degli Elaboratori II	inf/01	6	2	2
mfn0960	Modelli Concorrenti e Algoritmi distribuiti	inf/01	6	1 o 2	
mfn0795	Sistemi di Calcolo Paralleli e Distribuiti	inf/01	6	1 o 2	
Blocco 5	Tre insegnamenti a scelta tra: <i>(Nota: Tutti gli insegnamenti da 9 cfu presenti nei blocchi 2 e 3 sono erogati anche con un programma ridotto da 6 cfu. Di conseguenza, sono inseribili in questa parte del piano carriera solo se non già selezionati in precedenza)</i>				
Eventuali insegnamenti se non già selezionati nel blocco precedente					
mfn0969	Architettura degli Elaboratori II	inf/01	6	2	2
mfn0960	Modelli Concorrenti e Algoritmi distribuiti	inf/01	6	1 o 2	
mfn0795	Sistemi di Calcolo Paralleli e Distribuiti	inf/01	6	1 o 2	
Altri insegnamenti, esclusi i corrispondenti insegnamenti erogati in versione da 9 cfu già selezionati in precedenza					
mfn0997	Algoritmi e Complessità – Parte A	inf/01	6	1 o 2	
inf0010	Apprendimento Automatico e Analisi Intelligente dei Dati – Parte A ANNI ALTERNI 14-15 SI	inf/01	6	1	1
mfn0993	Basi di Dati Multimediali – Parte A 14-15 NON ATTIVO	inf/01	6	2	2
mfn0944	Laboratorio Avanzato di Basi Dati ANNI ALTERNI 14-15 NO	inf/01	6	2	2
mfn0953	Modelli e Metodi per il Supporto alle Decisioni ANNI ALTERNI 14-15 NO	inf/01	6	2	2
inf0008	Programmazione per Dispositivi Mobili	inf/01	6	2	2
mfn0954	Reti Complesse	inf/01	6	1 o 2	
mfn0977	Reti Neurali	inf/01	6	1 o 2	
mfn0995	Sviluppo Software per Componenti e Servizi Web – Parte A	inf/01	6	1 o 2	2
Inf0011	Teoria dell'Informazione – Parte A	inf/01	6	1 o 2	1

mfn1361	Valutazione delle prestazioni: Simulazione e Modelli - Parte A	inf/01	6	1 o 2	1
mfn1360	Verifica dei Programmi Concorrenti - Parte A	inf/01	6	1 o 2	1
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.					
mfn0939	Calcolabilità e Complessità B	inf/01	6	1 o 2	
mfn0985	Istituzioni di Linguaggi formali	inf/01	6	1 o 2	
mfn0987	Istituzioni di Sistemi Intelligenti	inf/01	6	1 o 2	
mfn0989	Istituzioni di Sviluppo Software	inf/01	6	1 o 2	
mfn0990	Istituzioni di Tecnologie Web	inf/01	6	1 o 2	
mfn0945	Sicurezza I	inf/01	6	1 o 2	
Formazione Affine ed interdisciplinare:					
Due insegnamenti a scelta tra:					
mfn0971	Complementi di Analisi e Probabilità	3 mat/05, 3 mat/06	6	1 o 2	1
mfn0970	Fisica per Applicazioni di Realtà Virtuale	fis/01	6	1	1
mfn1001	Istituzioni di Economia e Gestione dell'Impresa	secs-p/08	6	1	1
mfn0984	Istituzioni di Logica	mat/01	6	1 o 2	
mfn0962	Metodi Numerici	mat/08	6	1 o 2	
mfn1349	Ottimizzazione Combinatoria ANNI ALTERNI 14-15 SI	mat/09	6	1	1
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	L- Lin/12	3	1 o 2	
mfn0979	Prova Finale		24	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 studi 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, 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 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 studi 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 corsi (alcuni comuni anche ad altri indirizzi) che permettono di seguire un percorso coerente.

Il curriculum offre dei gradi di libertà al suo interno in quanto tra i corsi 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 corsi caratterizzanti per accentuare la preparazione verso una tematica specifica.

Il curriculum prevede anche il completamento della formazione con corsi affini ed interdisciplinari (in area economica/gestionale 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 2014

Codice	Insegnamento	SSD	CFU	Anno	Anno consigliato
CURRICULUM Sistemi per il Trattamento dell'Informazione -- percorso 104					
Formazione caratterizzante:					
mfn0942	Intelligenza Artificiale e Laboratorio	inf/01	9	1	1
mfn0940	Modelli Avanzati e Architetture di Basi di Dati	inf/01	9	1	1
mfn0941	Sviluppo Software per Componenti e Servizi Web	inf/01	9	1 o 2	2

Due insegnamenti a scelta tra:					
mfn0943	Apprendimento Automatico e Analisi Intelligente dei Dati ANNI ALTERNI 14-15 SI	inf/01	9	1 o 2	1
mfn0947	Basi di Dati Multimediali 14-15 NON ATTIVO	inf/01	9	2	2
mfn0972	Elaborazione di Immagini e Visione Artificiale	inf/01	9	1 o 2	2
mfn0949	Sistemi Cognitivi	inf/01	9	1 o 2	2
mfn0948	Trattamento dei Dati e dei Processi Aziendali ANNI ALTERNI 14-15 SI	inf/01	9	1	1
Quattro insegnamenti a scelta tra:					
mfn1348	Agenti Intelligenti	inf/01	6	1 o 2	2
mfn0997	Algoritmi e Complessità – Parte A	inf/01	6	1 o 2	
mfn0969	Architettura degli Elaboratori II	inf/01	6	1 o 2	
mfn0951	Bioinformatica	inf/01	6	1 o 2	
mfn0944	Laboratorio Avanzato di Basi Dati ANNI ALTERNI 14-15 NO	inf/01	6	2	2
mfn0960	Modelli Concorrenti e Algoritmi distribuiti	inf/01	6	1 o 2	1
mfn0953	Modelli e Metodi per il Supporto alle Decisioni ANNI ALTERNI 14-15 NO	inf/01	6	2	2
inf0008	Programmazione per Dispositivi Mobili	inf/01	6	2	2
mfn0954	Reti Complesse	inf/01	6	1 o 2	2
mfn0952	Sicurezza II	inf/01	6	1 o 2	
<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	inf/01	6	1 o 2	
mfn0986	Istituzioni di Interazione Uomo Macchina	inf/01	6	1	1
mfn0988	Istituzioni di programmazione distribuita in rete	inf/01	6	1	1
mfn0987	Istituzioni di Sistemi Intelligenti	inf/01	6	1	1
mfn0989	Istituzioni di Sviluppo Software	inf/01	6	1	1
mfn0990	Istituzioni di Tecnologie Web	inf/01	6	1	1
mfn0945	Sicurezza I	inf/01	6	1 o 2	
Formazione Affine ed interdisciplinare:					
Due insegnamenti a scelta tra:					
mfn0971	Complementi di Analisi e Probabilità	3 mat/05, 3 mat/06	6	1 o 2	
mfn0946	Economia e Gestione delle Imprese Net Based	secs-p/08	6	1 o 2	1
mfn0970	Fisica per Applicazioni di Realtà Virtuale	fis/01	6	1	1
mfn0962	Metodi Numerici	mat/08	6	1	1
mfn1349	Ottimizzazione Combinatoria ANNI ALTERNI 14-15 SI	mat/09	6	1	1
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	L- Lin/12	3	1 o 2	
mfn0979	Prova Finale		24	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 Ricerca in Informatica presso il Dipartimento di Informatica

Il Dottorato di Ricerca in Informatica è ad ampio spettro e copre le seguenti macro-aree:

- Basi dati, sistemi informativi e ingegneria del software
- Elaborazione di segnali ed immagini
- Fondamenti dell'informatica e linguaggi di programmazione
- Intelligenza artificiale e interazione uomo-macchina
- Sistemi di elaborazioni e reti

Poiché il Dottorato intende fornire ai dottorandi una preparazione spendibile non solo nel settore accademico, ma anche presso enti di ricerca (pubblici e privati) e presso aziende tecnologicamente avanzate, esso favorisce il coinvolgimento dei dottorandi nei progetti di ricerca e sviluppo svolti in collaborazione con enti industriali, autorizzando anche stage presso tali enti. Il Dottorato incoraggia inoltre i dottorandi a collaborare attivamente ai progetti di ricerca di interesse nazionale ed internazionale a cui partecipano le strutture scientifiche afferenti al Dottorato con particolare attenzione alle iniziative promosse dalla Unione Europea.

Al Dottorato si accede attraverso esame di ammissione; Il bando prevede la possibilità di iscrizione 'sub condicione' all'esame di ammissione al dottorato per i laureandi (laurea quinquennale o magistrale) 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 nelle aree scientifiche di interesse del dottorato. 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 dell'indirizzo Informatica del dottorato di ricerca è la prof.ssa Mariangiola Dezani (dezani@di.unito.it). Informazioni dettagliate ed aggiornate sulla attività della Scuola di Dottorato in Scienza ed Alta Tecnologia (compreso il curriculum di Informatica) 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 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

University of Helsinki, Finlandia

Universite Jean Monnet de Saint-Etienne, Francia

Sabaci University, Istanbul, Turchia

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 raccoglie tutti i libri di testo (spesso in più copie) e il materiale di riferimento (manuali, testi applicativi, dispense, ecc.) dei diversi corsi. Molti corsi, infatti, consigliano, accanto ad un testo fondamentale o a una raccolta di dispense,

altri libri complementari: la biblioteca fornisce così la possibilità di consultare tali testi e, in certi limiti, di ottenerne il prestito per lo studio individuale.

Per quanto riguarda l'aspetto di ricerca la biblioteca possiede molti volumi specialistici e una vasta raccolta di riviste scientifiche. La Biblioteca, attualmente, è aperta tutti i giorni, e i libri e le riviste possono essere consultati da tutti gli studenti dietro richiesta al personale. Un catalogo elettronico consente anche ricerche rapide di materiale bibliografico.

Supporto on-line ai corsi

Oltre ai libri di testo e ad eventuali dispense, i docenti dei corsi 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 corso.

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 dei corsi 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 alla 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 agli 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 Studi 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 Studi è la prof.ssa Cristina Baroglio.

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. 0116707611, cristina.baroglio@unito.it, Dipartimento di Informatica, Via Pessinetto 12, Torino

Supporto agli 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 – Fax 011.670.4285 – Email: sportello.dislessia@unito.it

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

Indirizzi utili

Per avere ulteriori e più recenti informazioni sull'organizzazione didattica è possibile accedere al sito:
<http://www.informatica.unito.it>

Ufficio del Corso di Studi 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 Studio in Informatica: prof.ssa Susanna Donatelli

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

Commissione ammissioni (verifica dei requisiti di ingresso), responsabile il prof. Piero Torasso

e-mail: ammissione-lm18@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

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

Tel. 011 – 670 4952 oppure 011 – 670 4953 E_mail: tasse.rimborsi@unito.it

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 dei Corsi per l’A.A. 14/15

In questa seconda parte della guida vengono riportati tutti gli insegnamenti attivati nell’anno accademico 14/15 presso il Corso di Studi di Informatica, di interesse per gli studenti immatricolati nell’anno e per gli studenti della coorte precedente che devono seguire i corsi del secondo anno. Sono inoltre riportati i corsi che sono tenuti in mutuaione dalla triennale, per il recupero di competenze. Infine, per facilitare la compilazione del piano carriera degli studenti della coorte 2014, questa seconda parte della guida riporta anche un elenco dei corsi che non verranno tenuti nel 14/15 ma che saranno attivi nel 15/16 (vuoi perché sono insegnamenti previsti ad anni alterni, vuoi perché si tratta di corsi di nuova istituzione l’attivazione dei quali è già stata deliberata).

Corsi attivi nel 14/15.

La tabella che segue elenca tutti i corsi che potrete seguire nell’A.A. 14/15. Se un corso elencato nell’offerta formativa della vostra coorte non è presente in questa tabella, allora vuol dire che il corso sarà attivo nel prossimo A.A., vuoi perché si tratta di un corso di nuova attivazione che è previsto al secondo anno della vostra coorte (e quindi nel 14/15 nessun studente potrebbe averlo nel carico didattico), vuoi perché si tratta di un esame 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 il corso. I programmi dei corsi sono riportati in appendice. Per ulteriori informazioni potete visitare le pagine web del corso di laurea (da <http://www.informatica.unito.it> seguire il link ‘corsi’, nella sezione “Informazioni per gli studenti”).

Codice principale	INSEGNAMENTO	SSD	CFU	DOCENTI	Semestre	NOTE
mfn1348	Agenti Intelligenti	inf/01	6	Martelli, Baldoni	2	
mfn0997	Algoritmi e Complessità A	inf/01	6	Zacchi	2	
mfn0943	Apprendimento Automatico e Analisi Intelligente dei Dati	inf/01	9	Meo, Esposito	2	Ad anni alterni, si terrà nel 14-15
mfn0969	Architettura degli Elaboratori II	inf/01	6	Gunetti	1	
mfn0951	Bioinformatica	inf/01	6	Botta Cordero	1	
mfn0939	Calcolabilità e Complessità B	inf/01	6	Ronchi della Rocca	2	mutuato su Calcolabilità e Complessità A, triennale 270; per il 14-15 mutuato su Metodi Formali dell’Informatica, triennale 270.
mfn0971	Complementi di Analisi e Probabilità	3 mat/05, 3 mat/06	6	Bibbona/Boggiatto	1	
mfn0946	Economia e Gestione delle Imprese Net Based	secs-p/08	6	Pironti/Pisano	2	
mfn0972	Elaborazione di Immagini e Visione Artificiale	inf/01	9	Balossino, Cavagnino, Grangetto	1	
mfn0974	Elaborazione Digitale Audio e Musica	inf/01	6	Lombardo, Valle	2	mutuato su corsi presso DAMS (Informazioni alla pagina: http://www.di.unito.it).

						it/~vincenzo/ElabSu oMus (corso tenuto presso DAMS-Palazzo Nuovo, Via Verdi Torino)
mfn0970	Fisica per Applicazioni di Realtà Virtuale	fis/01	6	Maggiara	1	
mfn0942	Intelligenza Artificiale e Laboratorio	inf/01	9	Martelli, Torasso	2	
mfn0963	Lingua Inglese II	L-Lin/12	3	Griffin (esercitatore), Radicioni, Cordero, Zacchi responsabili	1 e 2	
mfn0962	Metodi Numerici	mat/08	6	Semplice	1	
mfn0973	Modellazione Grafica	inf/01	9	Balossino, Cavagnino, Faletti	2	
mfn0940	Modelli Avanzati e Architetture di Basi di Dati	inf/01	9	Sacco	1	
mfn0960	Modelli Concorrenti e Algoritmi distribuiti	inf/01	6	Margaria	1	
mfn1349	Ottimizzazione Combinatoria	mat/09	6	Aringhieri, Grosso	2	Ad anni alterni, si terrà nel 14-15
mfn0954	Reti Complesse	inf/01	6	Ruffo, Sereno	1	Dal 2015-16 sarà mutuato da modulo di Analisi e Visualizzazione di Reti Complesse
mfn0968	Reti II	inf/01	9	Ruffo, Garetto	1	Corso riservato alla coorte 2013 o precedenti. Ultimo anno in cui viene tenuto
inf0009	Reti II	inf/01	6	Garetto	1	
mfn0977	Reti Neurali	inf/01	6	Cancelliere, Gliozzi	1	
mfn0952	Sicurezza II	inf/01	6	Bergadano	1	
mfn0949	Sistemi Cognitivi	inf/01	9	Boella, Mazzei, Radicioni	2	
mfn0795	Sistemi di Calcolo Paralleli e Distribuiti	inf/01	6	Aldinucci	2	mutuato su corso presso Laurea Magistrale in Fisica. Informazioni alla pagina http://fisica.campusnet.unito.it/do/corsi.pl/Show?_id=48d4;sort=DEFAULT;search=;hits=126 - http://fisica.campusnet.unito.it/do/lezioni.pl
mfn0978	Sistemi di Realtà Virtuale	inf/01	9	Balossino, Grangetto, Lucenteforte	1	
mfn0941	Sviluppo Software per Componenti e Servizi Web	inf/01	9	Petrone	1	
mfn0961	Teoria dell'Informazione	inf/01	9	Grangetto, Sereno	2	
mfn0948	Trattamento dei Dati e dei Processi Aziendali	inf/01	9	Di Leva	2	
mfn0899	Valutazione delle prestazioni: Simulazione e Modelli	inf/01	9	Balbo	2	
mfn0959	Verifica dei Programmi Concorrenti	inf/01	9	Donatelli	1	
mfn0979	Prova finale	altre attività	24			

Corsi mutuati attivi nell'A.A. 14/15:

I corsi mutuati sono corsi con un codice proprio, ma per i quali non è previsto un insegnamento proprio, bensì una mutuazione da tutto o parte di un altro corso. Vengono inseriti con due scopi principali: recupero di competenze utili della triennale oppure versione “corta”, cioè da 6 cfu, di un corso da 9 cfu della magistrale. Nel primo caso i corsi tipicamente si chiamano “istituzioni di XX” e mutuanano sul corso di nome “XX” della 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 mutuanano 6 dei 9 cfu del corso “XX”) e possono essere inseriti in modo autonomo dallo studente in accordo con le regole dei piani carriera, normalmente nel gruppo dei corsi liberi o in precisi gruppi di scelta, a patto che nel piano carriera dello studente non sia già presente la versione da 9 cfu.

Codice principale	INSEGNAMENTO	SSD	CFU	CORSO DA CUI MUTUA
Corsi del primo tipo				
mfn0939	Calcolabilità e Complessità B	inf/01	6	mutuato su Calcolabilità e Complessità A, triennale 270; per il 14-15 mutuato su Metodi Formali dell'Informatica, triennale 270.
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	1 2	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
Corsi del secondo tipo				
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
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

Corsi ad anni alterni non tenuti nel 14/15 e corsi di futura attivazione:

Nell'A.A. 15/16, in coincidenza del secondo anno della coorte 2014, saranno attivati i seguenti corsi:

- Analisi e Visualizzazione di Reti Complesse, codice inf0007, nel SSD inf/01
- Basi di Dati Multimediali, codice mfn0947, nel SSD inf/01
- Gestione delle Reti, codice inf0006, nel SSD inf/01
- Laboratorio Avanzato di Basi Dati, codice mfn0940, nel SSD inf/01
- Modelli e Metodi per il Supporto alle Decisioni, codice mfn0953, nel SSD inf/01
- Programmazione per Dispositivi Mobili, codice inf0008, nel SSD inf/01

I programmi previsti sono i seguenti (fra parentesi i nomi dei docenti che hanno contribuito alla definizione del programma)

Analisi e Visualizzazione di Reti Complesse (prof. Ruffo, prof. Sereno)

- Introduzione alle reti complesse: esempi dalla fisica, biologia, sociologia, economia, spettacolo ed informatica.

- Esempi: caratteristiche di Internet, network degli attori, etc.

Ripasso di teoria dei grafi. Definizione di grafo. Grafi semplici e bipartiti. Grafi pesati e non. Grafi diretti e non.

- Caratterizzazione della topologia di un network a livello globale e di singolo nodo: distribuzione dei parametri per singolo nodo (connettività). Connettività, clustering, misure di centralità.

- Diametro di un network. Sottonetworks; clustering di un network; cliques e moduli. Definizione e calcolo delle principali misure del network. Algoritmi per l'identificazione di comunità.

- Modello base: random networks alla Erdos-Renyi. Distribuzione dei parametri del network e teoremi limite per reti di grandi dimensioni.

Matrici di Wigner e spettro degli autovalori. Transizione di fase e giant cluster. Relazione tra vari parametri del network (assortativity, connettività vs. betweenness centrality).

- Generalizzazione a small world networks: high clustering short distances (log(N)). Modello di Watts-Strogatz: rewiring.

- Scale free networks: esempi. Modello di crescita di Barabasi-Alberts: preferential attachment. Scale log-log. Istogrammi e possibili errori (heavy tails).
- Casi di Studio: Social Media, Sistemi P2P; Reti di similarita' ed applicazioni nei sistemi di recommendation; fenomeni ed algoritmi epidemici; Applicazioni di reti sociali e comunita' virtuali.
- Visualizzazione di dati complessi (Metodi): Algoritmi per la visualizzazione di grafi; tassonomia dei layout; force based layout; tree layout; circular layout; geography layout; vantaggi e svantaggi nell'uso delle mappe.
- Visualizzazione di dati complessi (Strumenti): Gephi; librerie igraph/networkx per Python/R; librerie per il Web: d3.js, Google charts e Google Maps; Rappresentazione dati e tecnologie NoSQL: fondamenti di Neo4J e MongoDB.

Basi di Dati Multimediali (prof. Sapino)

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 Università straniere e di ricercatori presso centri di ricerca dell'area torinese.

Gestione delle Reti (prof. Manini)

- Protocolli di routing: Nuclei, Paritari, e algoritmi
- Protocolli per router esterni e sistemi autonomi (BGP)
- Routing all'interno di un sistemi autonomo (RIP, OSPF, HELLO)
- 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;

Laboratorio Avanzato di Basi Dati (prof. Meo)

- 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 integrità 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.
- Data Warehouses: schema a stella, Dimensional Fact Model, dimensioni e gerarchie sulle dimensioni, dimensioni variabili nel tempo
- Strumenti di OLAP e Business Intelligence (i cosiddetti *cruscotti decisionali*)

Modelli e Metodi per il Supporto alle Decisioni (prof. Aringhieri)

1. Introduzione. Richiami ed elementi di base di Simulazione e di Ottimizzazione.
2. Modelli per descrivere l'iterazione tra componenti diverse di un sistema: Agent-based simulation.
3. Modelli per l'analisi di politiche di gestione di medio e lungo termine: System Dynamics.
4. Algoritmi euristici di ottimizzazione.
5. Integrazione tra simulazione ed ottimizzazione: case studies.

Programmazione per Dispositivi Mobili (prof. Ruffo)

- Introduzione allo sviluppo di applicazioni per dispositivi mobili
- Elementi di Interaction Design e Usabilità per app
- Richiami del pattern MVC
- Fondamenti del linguaggio Objective-C
- Architetture ARM e Intel
- Panoramica dei Sistemi Operativi per dispositivi mobili
- Pubblicazione delle app negli "Store"
- Modelli di business e monetizzazione
- Sviluppo di app native per Android e iOS
- Uso di Eclipse/XCode
- Navigazione e progettazione dell'interfaccia
- Persistenza dei dati
- Uso delle liste per la navigazione e la visualizzazione dei dati
- Mappe e localizzazione
- Accesso alle risorse del dispositivo e ai sensori
- Sviluppo di applicazioni multi-piattaforma e per il Web
- HTML5/CSS3/JS
- PhoneGap e applicazioni ibride
- Elementi per lo sviluppo con Windows phone

Corsi esterni all'Ateneo.

Il Corso di Studi 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* ai corsi scelti come corsi liberi – almeno 12 cfu). Solo in questo modo sarà possibile effettuare l'eventuale convalida: al momento della domanda di laurea, ciò che non 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 a Luglio 2014

Programmi e altre informazioni per i corsi attivi nel 14/15 (syllabus dei corsi)

Syllabus

N° documenti: 40

Testi del Syllabus

Docente	BALDONI MATTEO	Matricola: 013943
Anno offerta:	2014/2015	
Insegnamento:	MFN1348 - AGENTI INTELLIGENTI (Intelligent agents)	
Corso di studio:	008515 - INFORMATICA (Computer science)	
Anno regolamento:	2014	
CFU:	6	
Settore:	INF/01	
Tipo attività:	B - Caratterizzante	
Partizione studenti:	-	
Periodo:	Secondo Semestre	



Testi in italiano

Tipo testo	Testo
Lingua insegnamento	Italiano
Contenuti	<ul style="list-style-type: none">- Introduzione agli agenti. Proprietà. Diverse visioni.- Modello generale di agente. Agenti intelligenti. Agenti e oggetti. Agenti come sistemi intenzionali. Architettura astratta di agenti: agenti reattivi e con stato.- Agenti deduttivi.- Practical reasoning agents. Ciclo di controllo di practical reasoning agents.- Logiche per sistemi multiagente: logica modale, dinamica, temporale. Logiche per knowledge e belief. Ragionamento su azioni.- Modelli formali di agenti: la logica di Cohen e Levesque.- Linguaggi per agenti: PRS, Agent Oriented Programming, Concurrent METATEM, GOLOG e DYLOG.- Agenti reattivi e ibridi: la Subsumption Architecture di Brooks, il Mars explorer di Steels, architetture ibride, Touring Machines.- Multiagent Systems.- Comunicazione fra agenti: Speech acts come azioni. Agent Communication Languages (ACL): KQML e FIPA. Protocolli di interazione. Contract Net. Linguaggi di coordinazione. Semantica sociale di ACL.- Interazioni fra agenti: teoria dei giochi, strategie e equilibri di Nash, il dilemma del prigioniero, giochi iterati.- Come raggiungere un accordo: Protocolli e strategie. Aste. Negoziazione in Task Oriented Domains.- Introduzione a Jade.- Introduzione a Jason.
Testi di riferimento	<p>Testo Principale:</p> <p>Michael Wooldridge, An Introduction to MultiAgent Systems. John Wiley & Sons, 2002.</p> <p>Altri:</p> <ul style="list-style-type: none">- 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.
Obiettivi formativi	Il corso ha l'obiettivo di introdurre lo studente alla tecnologia ad agenti e ai sistemi multi-agente.
Prerequisiti	Principi metodologici dell'Intelligenza Artificiale, Programmazione e Algoritmi.
Modalità di verifica dell'apprendimento	Esame orale.
Risultati di apprendimento attesi	Alla fine del corso lo studente avrà acquisito le nozioni fondamentali su agenti e sistema multiagent.
Frequenza	Non obbligatoria ma consigliata.



Testi in inglese

Tipo testo	Testo
Lingua insegnamento	Italian
Contenuti	<ul style="list-style-type: none">- Introduction to agents. Properties.- Agent model. Intelligent agents. Agents vs objects. Intentions and agents. Abstract agent architecture: reactive agents and agents with internal state.- Deductive agents.- Practical reasoning agents. Control cycle.- Formalism for agent modeling: Cohen and Levesque's logic.- Agent languages: PRS, Agent Oriented Programming, Concurrent METATEM, GOLOG and DYLOG.- Reactive and hybrid agents: Brooks' subsumption architecture, Steels' Mars explorer, hybrid architecture, Touring Machines.- Multiagent systems.- Agent communications: speech acts. Agent communication languages (ACL): KQML and FIPA. Interaction protocols. Contract Net. Coordination languages. Social semantics for ACL.- Agent interaction: game theory, strategy and Nash equilibria, prisoner dilemma, iterated games.- Agreements: protocols and strategies. Auctions. Negotiation.- Introduction to Jade and Jason.
Testi di riferimento	<p>Main:</p> <p>Michael Wooldridge, An Introduction to MultiAgent Systems. John Wiley & Sons, 2002.</p> <p>Others:</p> <ul style="list-style-type: none">- 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.
Obiettivi formativi	The aim is to introduce to the agent technology and multiagent systems.
Prerequisiti	Artificial Intelligence, Programming and Algorithms.
Modalità di verifica dell'apprendimento	Oral exam.
Risultati di apprendimento attesi	At the end of the course the student has acquired the fundamental notions about agent and multiagent systems.
Frequenza	Attendance is recommended.

Testi del Syllabus

Docente	ZACCHI MADDALENA	Matricola: 011333
Anno offerta:	2014/2015	
Insegnamento:	MFN0997 - ALGORITMI E COMPLESSITA' - PARTE A (Algorithms and Complexity - Part A)	
Corso di studio:	008515 - INFORMATICA (Computer science)	
Anno regolamento:	2014	
CFU:	6	
Settore:	INF/01	
Tipo attività:	B - Caratterizzante	
Partizione studenti:	-	
Periodo:	Secondo Semestre	



Testi in italiano

Tipo testo	Testo
Lingua insegnamento	Italiano
Contenuti	<p>Programmazione dinamica. Uso "top-down" e "bottom-up" di definizioni ricorsive di funzioni. Esempi: Zaino, Cammini minimi nei grafi, Prodotto di matrici.</p> <p>Backtracking e Branch and Bound. Esempi: Il problema delle n regine, Zaino, Commesso Viaggiatore.</p> <p>Complessità polinomiale in tempo. Certificati polinomiali, algoritmi non deterministici, le classi P e NP. Riducibilità polinomiale, NP-completezza.</p> <p>Algoritmi di approssimazione. - Approssimazione assoluta ed errore relativo. - Schemi di approssimazione pienamente polinomiali.</p> <p>Algoritmi randomizzati e algoritmi euristici (introduzione).</p>
Testi di riferimento	<p>T. H.Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, "Introduzione agli algoritmi e strutture dati", Seconda edizione, McGraw-Hill, 2005.</p> <p>A. Bertossi, "Algoritmi e strutture di dati", UTET Libreria, 2000.</p> <p>C.H. Papadimitriou, "Computational Complexity", Addison-Wesley, Longman, 1995.</p> <p>R. Sedgwick, "Algoritmi in C", Addison-Wesley, Masson, 1993.</p> <p>J. Kleinberg, E. Tardos, "Computer algorithms", Pearson Education, 2006.</p>
Obiettivi formativi	<p>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.</p>
Prerequisiti	<p>Programmazione dinamica. Uso "top-down" e "bottom-up" di definizioni ricorsive di funzioni. Esempi: Zaino, Cammini minimi nei grafi, Prodotto di matrici.</p> <p>Backtracking e Branch and Bound. Esempi: Il problema delle n regine, Zaino, Commesso Viaggiatore.</p> <p>Complessità polinomiale in tempo. Certificati polinomiali, algoritmi non deterministici, le classi P e NP. Riducibilità polinomiale, NP-completezza.</p> <p>Algoritmi di approssimazione. Approssimazione assoluta ed errore relativo. Schemi di approssimazione pienamente polinomiali.</p> <p>Algoritmi randomizzati e algoritmi euristici (introduzione).</p>
Propedeutico a	Nessuna propedeuticità obbligatoria.
Modalità di verifica dell'apprendimento	Esame scritto e presentazione di un seminario di approfondimento su uno degli argomenti del corso.
Risultati di apprendimento attesi	<p>La tesi di Church-Turing sulla calcolabilità va riveduta 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.</p>



Testi in inglese

Tipo testo

Testo

Lingua insegnamento

Italian

Contenuti

Dynamic Programming. Use "top-down" and "bottom-up" of recursive definitions of functions. Examples: Knapsack, shortest paths in graphs, Product of matrices.
Backtracking and Branch and Bound. Examples: The problem of the n queens, Knapsack, Salesman problem.
Complexity in polynomial time. Polynomial certificates, non-deterministic algorithms, classes P and NP. Polynomial reducibility, NP-completeness.
Approximation algorithms. - Absolute approximation and relative error. - Fully polynomial approximation schemes.
Randomized algorithms and heuristic algorithms (an introduction).

Testi di riferimento

T. H.Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, "Introduction to Algorithms", MIT, 2001.
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.
J. Kleinberg, E. Tardos, "Computer algorithms", Pearson Education, 2006.

Obiettivi formativi

The course is designed to provide advanced knowledge for the design, analysis and comparison of algorithms. A further aim of the course is to make known programming techniques that are designed to address problems that are intractable, possibly providing approximate solutions or answers only probably correct.

Prerequisiti

Dynamic Programming. Use "top-down" and "bottom-up" of recursive definitions of functions. Examples: knapsack, shortest paths in graphs, product of matrices.
Backtracking and Branch and Bound. Examples: the problem of the n queens, knapsack, the salesman problem.
Complexity in polynomial time. Polynomial certificates, non-deterministic algorithms, classes P and NP. Polynomial-time reductions, NP-complete problems.
Approximation algorithms. Absolute approximation and relative error. Fully polynomial approximation schemes.
Randomized algorithms and heuristic algorithms (an introduction).

Propedeutico a

It is not a mandatory prerequisite to any course.

Modalità di verifica dell'apprendimento

Written test and presentation of a seminar on one of the topics of the course.

Risultati di apprendimento attesi

The Church-Turing thesis on computability must be reviewed in reference to the actual costs which the computations require. The student should be able to evaluate the efficiency of algorithms and, if they require huge costs in practice, address the problem of finding programs quick and inexpensive, however, that provide acceptable solutions.

Testi del Syllabus

Docente	MEO ROSA	Matricola: 013995
Anno offerta:	2014/2015	
Insegnamento:	MFN0943 - APPRENDIMENTO AUTOMATICO E ANALISI INTELLIGENTE DEI DATI (Machine Learning and Intelligent Data Analysis)	
Corso di studio:	008515 - INFORMATICA (Computer science)	
Anno regolamento:	2014	
CFU:	9	
Settore:	INF/01	
Tipo attività:	B - Caratterizzante	
Partizione studenti:	-	
Periodo:	Secondo Semestre	



Testi in italiano

Tipo testo

Testo

Lingua insegnamento

italiano

Contenuti

Il corso copre le basi per comprendere e affrontare i problemi dell'Apprendimento Automatico e del Data Mining e più in generale le problematiche relative al processo di estrazione della conoscenza da basi di dati (KDD).
Si studiano le tecniche di preparazione dei dati: riduzione delle dimensioni, campionamento.
Si richiamano i principi di statistica: distribuzione delle statistiche campionarie, statistica inferenziale, stima parametrica, test di indipendenza.
Si discutono i problemi legati all'elevata dimensionalità e alcune metodologie di selezione delle dimensioni.
Si discutono alcune tecniche di descrizione dei concetti: caratterizzazione e classificazione.
Si presentano alcuni dei principali algoritmi di classificazione (alberi decisionali, K-nearest-neighbors, a regole, reti neurali, Support Vector Machines).
Si richiamano le tecniche di regressione.
Si discute l'estrazione di pattern frequenti e delle associazioni (tra variabili e itemset).
Si mostrano alcuni dei principali algoritmi di estrazione di regole di associazione: Apriori, FP-Growth, con campionamento.
Analisi dei dati tramite Clustering.
Algoritmi di clustering: K-means, density-based, probabilistici, concettuali.
Validazione e verifica dei risultati sui dati.
Sperimentazione con un sistema reale di analisi dei dati su vari data set: Weka, RapidMiner, R.

Testi di riferimento

Introduction to Data Mining, by Pang-Ning Tan, Michael Steinbach, Vipin Kumar, (2nd edition), Addison Wesley, 2006.

Obiettivi formativi

Oltre all'esposizione delle principali tematiche che caratterizzano il Data Mining (classificazione, clustering, estrazione di pattern ricorrenti) si vuole fornire allo studente una base di conoscenza di statistica e di teoria dell'informazione necessari per capire e utilizzare i fondamenti teorici di questi approcci.

Inoltre, si vogliono proporre alcune tecniche per affrontare l'analisi dei dati su grandi volumi di dati, quali le tecniche di pre-processing (pulizia dei dati, il campionamento, selezione delle dimensioni) e la validazione sui dati dei risultati dell'analisi e la loro interpretazione.

Infine si vuole proporre allo studente uno strumento di data mining quale ad esempio RapidMiner, KNIME, Weka o R, e alcune esperienze pratiche. In particolare si sperimenteranno alcuni algoritmi di analisi dei dati e apprendimento automatico disponibili pubblicamente e già implementati in Java (Suite di algoritmi WEKA).

Prerequisiti

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

Propedeutico a

Nessuno.

Tipo testo

Testo

Note

In caso lo studente prenda questo corso in un formato da 6 CFU (e non da 9) potrà scegliere di studiare uno dei seguenti temi in alternativa: classificazione, clustering, e pattern frequenti e associazioni.

Modalità di verifica dell'apprendimento

La verifica dell'apprendimento avviene sia in itinere che a fine corso in sede di esame.

La verifica in itinere si svolge durante le esercitazioni, in cui lo studente interagisce direttamente con un software di analisi dati su dei data-set reali.

La verifica a fine corso consiste nell'esame (orale) durante il quale lo studente oltre a rispondere a domande di tipo teorico mostra le proprie competenze utilizzando il software e mostrando di saper analizzare un data-set già analizzato durante le esercitazioni del corso.

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 e sul cui utilizzo lo studente verrà anche valutato in sede di esame.

A titolo di esempio si includono domande che potrebbero essere fatte all'orale.

1. Descrivere le principali tecniche per la pulizia dei dati.
2. Illustrare la selezione delle dimensioni (attributi) in base alla rilevanza.
3. Illustrare in che cosa consiste la caratterizzazione di una classe.
4. Scrivere lo pseudo-codice di alcuni algoritmi di classificazione o clustering o di estrazione di pattern frequenti
5. Descrivere gli utilizzi e la struttura di una tavola di contingenza.
6. Descrivere le 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. Illustrare alcune delle misure di valutazione degli itemset
13. Discutere alcune misure di similarità e specificare per quali tipi di variabili sono applicabili
14. Descrivere e confrontare alcuni degli approcci visti a lezione per fare clustering.
15. Illustrare uno degli esercizi svolti in laboratorio con SAS/RapidMiner/Weka.

Risultati di apprendimento attesi

Conoscenze di statistica di base per l'analisi dei dati, quali analisi di serie storiche, test di indipendenza.

Acquisizione delle modalità in cui avviene il processo di KDD, e relativa scelta delle tecniche da adottare in ciascuna fase del processo.

Scelta della tecnica di analisi da adottare per un certo insieme di dati.

Analisi di un algoritmo di data mining.

Interpretazione e validazione dei risultati dell'analisi.

Frequenza

non obbligatoria ma suggerita



Testi in inglese

Tipo testo	Testo
Lingua insegnamento	italian
Contenuti	<p>This course introduces to the comprehension of the problems and of the main methodologies of machine learning and data mining. It discusses the process of knowledge discovery from data (KDD). It shows the primitives of Data Mining, languages, and architecture of the systems.</p> <p>It introduces the techniques for the preparation of data: reduction of the dimensions, sampling.</p> <p>It recalls the principles of statistics: sampling distribution of a statistics, inferential statistics, parametric estimate, independence test.</p> <p>It shows some of the problems linked to the high dimensionality and shows some of the algorithms for the selection of the dimensions.</p> <p>It presents the description of concepts: characterization and classification. It discusses some of the main algorithms of classification (decision trees, K-nearest-neighbors, classification rules, neural networks, Support Vector Machines).</p> <p>It recalls regression (linear and logistic).</p> <p>It treats the discovery of frequent patterns and of association of variables.</p> <p>It shows some of the main algorithms for the extraction of frequent patterns: Apriori, FP-Growth, with sampling.</p> <p>Analysis of data by means of Clustering.</p> <p>Clustering algorithms: K-means, density-based, probabilistic, conceptual.</p> <p>Validation and verification of the results on data.</p> <p>Experiments with a real system on data sets: Weka, RapidMiner, KNIME.</p>
Testi di riferimento	Introduction to Data Mining, by Pang-Ning Tan, Michael Steinbach, Vipin Kumar, (2nd edition), Addison Wesley, 2006.
Obiettivi formativi	<p>Besides the discussion of the main techniques that characterize Data Mining (classification, clustering, frequent pattern discovery) we aim at providing a basilar knowledge of statistics and information theory necessary to the understanding and deployment of the fundamentals of these approaches.</p> <p>Furthermore, we aim at presenting some techniques for the data analysis on big data, such as pre-processing (data cleaning, sampling, dimension selection) and the validation of the results of the data analysis and their interpretation.</p> <p>Finally we want to propose a practical tool for data mining such as RapidMiner, KNIME, R and some practical esperiences on some real data with a software suite in Java (WEKA).</p>
Prerequisiti	Intelligent systems and Databases.
Propedeutico a	None
Note	<p>In the case the student takes this course in a reduced form (6 CFU instead of 9) she/he will be allowed to choose between one of the following topics as alternatives: classification, clustering, frequent itemsets and associations.</p>

Tipo testo

Testo

Modalità di verifica dell'apprendimento

The test of the comprehension occurs in-progress (during the exercises in laboratory with the data analytics software) and during the oral test (with questions on the theory and some exercises already done during the laboratory).

For an exemplary purpose here there is a list of possible oral questions:

1. Describe the main techniques for data cleaning.
2. Present the dimensions selection on the basis of the attributes relevance.
3. Discuss the concept of class characterization.
4. Write the pseudo-code of some algorithms (classification, clustering or pattern extraction).
5. Describe the use and structure of a contingency table.
6. Describe the techniques for the validation of a classification method.
7. Describe the meaning and the use of the Chi-square statistics.
8. Discuss the meaning of an association rule.
9. Describe the search space of the large itemsets.
10. Describe the algorithm and the principle of Apriori.
11. Explain the meaning of the closed itemsets and their advantages.
12. Discuss some of the evaluation measures of the itemsets.
13. Discuss some of the measures of similarity and explain for which variables they are applicable.
14. Compare some of the approaches to do clustering.
15. Execute some of the experiments on data analysis, seen in laboratory with RapidMiner, Weka or R.

Risultati di apprendimento attesi

Knowledge on statistics, of the knowledge discovery process, and of the various steps.

Ability to choose the correct data analysis technique.

Analysis of a data mining algorithm. Interpretation and validation of the results.

Frequenza

non compulsory but suggested

Testi del Syllabus

Docente	GUNETTI DANIELE	Matricola: 013602
Anno offerta:	2014/2015	
Insegnamento:	MFN0969 - ARCHITETTURE DEGLI ELABORATORI II (Computer Architecture II)	
Corso di studio:	008515 - INFORMATICA (Computer science)	
Anno regolamento:	2013	
CFU:	6	
Settore:	INF/01	
Tipo attività:	B - Caratterizzante	
Partizione studenti:	-	
Periodo:	Primo Semestre	

Testi in italiano

Tipo testo	Testo
Lingua insegnamento	Italiano
Contenuti	<p>* PARTE I: o Concetti di base delle architetture RISC</p> <p>o Concetti di base del Pipelining</p> <p>o Instruction Level Parallelism</p> <p>o Instruction Level Parallelism</p> <p>o Concetti fondamentali di Caching</p> <p>* PARTE II: o Introduzione -- Multithreading</p> <p>o Architetture Multiprocessore</p> <p>o Architetture Multicomputer Processori vettoriali una nota storica</p> <p>MISCELLANEA: o Sistemi RAID o Cenni sulla programmazione in assembler dell'8088 o seminario sulle architetture grafiche</p>
Testi di riferimento	<p>Andrew Tanenbaum: Structured Computer Organization (fifth Edition) *</p> <p>D. Patterson & J. Hennessy: Struttura e Progetto dei Calcolatori. L'interfaccia Hardware-Software (2a ed. - terza ed. americana) *</p> <p>J. Hennessy & D. Patterson: Computer Architecture, A quantitative Approach (3rd Edition) *</p> <p>J. Hennessy & D. Patterson: Computer Architecture, A quantitative Approach (4th Edition)</p>
Obiettivi formativi	Fornire conoscenze approfondite sull'architettura e il funzionamento dei moderni computer, con particolare attenzione alle architetture parallele.
Prerequisiti	nessuno
Propedeutico a	nessun insegnamento
Note	nessuna
Modalità di verifica dell'apprendimento	esame orale
Risultati di apprendimento attesi	I corso è diviso in due parti: nella prima verranno descritti i principi di funzionamento delle moderne architetture RISC, con particolare attenzione alle tecniche di esecuzione delle istruzioni comunemente utilizzate per aumentare l'efficienza computazionale: Caching, Pipelining, Esecuzione delle istruzioni "Out of order", Branch Prediction, Esecuzione Speculativa, Multiple Issue, istruzioni Predicative. Nella seconda parte del corso verranno invece descritte le varie tipologie di architetture parallele: da quelle multi-processore a quelle multi-computer.
Frequenza	non obbligatoria



Testi in inglese

Tipo testo	Testo
Lingua insegnamento	Italian
Contenuti	<p>PART I:</p> <ul style="list-style-type: none">o Introduction to RISC architectures: basic conceptso Pipeliningo Instruction Level Parallelismo Instruction Level Parallelismo Caching <p>* PART II: o Introduction and Multithreading</p> <ul style="list-style-type: none">o Multiprocessor Architectureso Multicomputers Architectures Vector Processors a historical note <p>MISCELLANEA: o RAID systems o an introduction to the 8088 and to the RISC assembler o GPGPU</p>
Testi di riferimento	<p>Andrew Tanenbaum: Structured Computer Organization (fifth Edition) *</p> <p>D. Patterson & J. Hennessy: Struttura e Progetto dei Calcolatori. L'interfaccia Hardware-Software (2a ed. - terza ed. americana) *</p> <p>J. Hennessy & D. Patterson: Computer Architecture, A quantitative Approach (3rd Edition) *</p> <p>J. Hennessy & D. Patterson: Computer Architecture, A quantitative Approach (4th Edition)</p>
Obiettivi formativi	To provide an in depth understanding of modern computer architecture and functioning, with particular attention to parallel architectures
Prerequisiti	none
Propedeutico a	no teaching
Note	none
Modalità di verifica dell'apprendimento	oral examination
Risultati di apprendimento attesi	<p>This course is divided in two parts. In the first part we will describe the principle of functioning of modern RISC architectures, with particular attention to the techniques used to execute programs instructions so as to improve computational efficiency: caching, pipelining, out of order execution, branch prediction, speculative execution, multiple issue, predicative instructions.</p> <p>In the second part of the course we will describe the various kind of parallel architectures: from multiprocessors to multicomputers.</p>

Tipo testo**Testo****Frequenza**

not mandatory

Testi del Syllabus

Docente	BOTTA MARCO	Matricola: 013397
Anno offerta:	2014/2015	
Insegnamento:	MFN0951 - BIOINFORMATICA (Bioinformatics)	
Corso di studio:	008515 - INFORMATICA (Computer science)	
Anno regolamento:	2014	
CFU:	6	
Settore:	INF/01	
Tipo attività:	B - Caratterizzante	
Partizione studenti:	-	
Periodo:	Primo Semestre	



Testi in italiano

Tipo testo	Testo
Lingua insegnamento	Italiano
Contenuti	Introduzione alla biologia molecolare. Algoritmi di allineamento di sequenze. Algoritmi di allineamento multiplo. Costruzione di alberi filogenetici. Struttura delle proteine e predizione della struttura. Reti di regolazione genica: modelli e algoritmi di inferenza. Organizzazione e interrogazione di database biologici.
Testi di riferimento	A.M. Lesk, "Introduzione alla Bioinformatica", McGraw-Hill Ed. G. Valle, M. Helmer Citterich, M. Attimonelli, G. Pesole "Introduzione alla Bioinformatica", Zanichelli Ed. D.E. Krane e M.L. Raymer, "Fondamenti di Bioinformatica", Pearson Education Italia Ed.
Obiettivi formativi	Il corso fornisce, da un lato, i fondamenti della problematiche biologiche e, dall'altro, delle tecniche algoritmiche per affrontare questi problemi.
Prerequisiti	Formalmente nessuno. Nozioni di algoritmi, intelligenza artificiale e biologia di base possono essere utili.
Modalità di verifica dell'apprendimento	scritta e orale
Risultati di apprendimento attesi	Al termine del corso lo studente sarà in grado di colloquiare con il biologo, comprendere il problema biologico e suggerire una soluzione informatizzata basata sulle tecniche apprese nel corso.
Frequenza	non obbligatoria, ma consigliata



Testi in inglese

Tipo testo	Testo
Lingua insegnamento	Italian
Contenuti	Introduction to molecular biology. Sequence alignment algorithms. Multiple sequence alignment algorithms. Phylogenetic tree construction algorithms. Protein structure prediction algorithms. Gene networks: models and inference algorithms. Organization and query of biological databases.
Testi di riferimento	A.M. Lesk, "Introduzione alla Bioinformatica", McGraw-Hill Ed.
Obiettivi formativi	The course provides, on one side, a general overview of biological problems, and, on the other side, describes the main techniques used to deal with these problems.
Prerequisiti	Formally none. Knowledge of algorithms, artificial intelligence and basic biology would help.
Modalità di verifica dell'apprendimento	written and oral
Risultati di apprendimento attesi	The student will be able to discuss with the biologist, to understand the biological problem at hand, and to suggest a solution based on the techniques learned during the course.
Frequenza	not mandatory, but suggested

Testi del Syllabus

Docente	BIBBONA ENRICO	Matricola: 033529
Anno offerta:	2014/2015	
Insegnamento:	MFN0971B - COMPLEMENTI DI PROBABILITA' (Elements of Probability)	
Corso di studio:	008515 - INFORMATICA (Computer science)	
Anno regolamento:	2014	
CFU:	3	
Settore:	MAT/06	
Tipo attività:	C - Affine/Integrativa	
Partizione studenti:	-	
Anno corso:	1	
Periodo:	Primo Semestre	

Tipo testo**Testo****Lingua insegnamento**

Italiano

Contenuti

Variabili aleatorie congiuntamente Distribuite.
Calcolo di probabilità, attese e varianze mediante condizionamento.
Catene di Markov a tempo discreto.
Variabile aleatorie esponenziale e processo di Poisson.
Catene di Markov a tempo Continuo.
Cenni al moto Browniano.

Testi di riferimento

Ross S.M. Introduction to probability models. Academic Press, 2003.

Obiettivi formativi

Il corso si propone di sviluppare negli studenti le capacità necessarie per formulare modelli probabilistici di situazioni di interesse applicativo. Lo studio di processi stocastici e delle relative proprietà verrà finalizzata alla formulazione di modelli relativi a situazioni reali. Tra gli obiettivi del corso vi è lo sviluppo delle capacità necessarie per la formulazione e lo studio di semplici modelli probabilistici.

INDICATORI DI DUBLINO (in riferimento al Regolamento Didattico di Ateneo, descrittori europei del titolo di studio- "descrittori di Dublino", http://www.study-in-italy.it/php5/scheda_corso.php?ambiente=off&anno=2009&corso=1214968)

Conoscenza e capacità di comprensione Il corso, partendo dalle conoscenze di base relative al calcolo delle probabilità (obiettivo 3), introduce i primi concetti relativi ai processi stocastici, utili per lo sviluppo di semplici modelli stocastici e per il loro studio (obiettivo 17). Alcuni esercizi proposti, legati alla simulazione di modelli, richiedono la scrittura di semplici programmi con software scelti dallo studente in base alle sue competenze computazionali e informatiche (obiettivo 18). Il corso utilizza un testo in inglese, adottato in larghissimo numero di università nel mondo. Questa scelta favorisce l'abitudine alla lettura di letteratura matematica in lingua inglese.

Capacità di applicare conoscenza e comprensione Il corso presenta nozioni relative ai processi stocastici, evitando l'utilizzo della teoria della misura, non vengono quindi presentati alcuni risultati che richiederebbero mezzi astratti. Si considerano solo risultati trattabili con i mezzi a disposizione a uno studente triennale. Questo non impedisce allo studente di comprendere e impadronirsi di concetti di primaria importanza. lavorandoci in modo rigoroso e riuscendo a dimostrare autonomamente alcuni risultati simili a quelli discussi in aula (obiettivo 1). Il corso è orientato alla soluzione di problemi, che vengono regolarmente assegnati, corretti dal docente e poi discussi in aula. Tali esercizi sono parte integrante della prova d'esame (obiettivi 2,3,4). In alcuni casi la soluzione degli esercizi può avvenire anche con l'ausilio di strumenti computazionali o informatici (obiettivo 5)

Autonomia di giudizio Gli esercizi che vengono proposti possono venir risolti individualmente o in gruppo. Il confronto con i compagni di corso, nel lavoro a casa o durante le correzioni in aula, favorisce lo sviluppo di capacità logiche per riuscire a chiarire ai compagni le proprie soluzioni (obiettivo 1 e 4). Spesso gli esercizi proposti possono venir risolti in modi molto diversi. La presentazione di soluzioni di altri permette di sviluppare capacità di riconoscimento di errori in dimostrazioni distinguendo anche dimostrazioni corrette alternative (obiettivo 3) .

Abilità comunicative Le numerose discussioni sui diversi metodi per risolvere gli esercizi proposti consentono di migliorare le capacità di comunicazione (obiettivo 1 per la lingua italiana). Inoltre la formalizzazione in modelli di semplici realtà fisiche, informatiche o

Tipo testo

Testo

biologiche allena lo studente a rivolgersi a un pubblico non matematico, presentando risultati di studi matematici (obiettivo 2). Il corso utilizza un testo in lingua inglese, rendendo familiare per lo studente l'uso scientifico di tale lingua (obiettivo 3)

Capacità di apprendimento Il corso fornisce alcuni concetti di base della teoria dei processi stocastici che saranno utili a quanti proseguiranno per avere in mente semplici esempi che illustreranno concetti più astratti (obiettivo 1). Gli stessi concetti, specie quelli relativi alla modellizzazione, potranno essere di estrema utilità in ambito lavorativo (obiettivo 2). L'apprendimento del metodo scientifico alla base della formulazione di modelli matematici potrà poi rivelarsi utile, anche a distanza di tempo, per la formalizzazione logica o matematica di realtà di svariata natura (obiettivo 4).

Modalità di verifica dell'apprendimento

Orale durante il quale si propone anche la soluzione di un esercizio. L'esercizio orale può essere sostituito dalla soluzione individuale o a gruppi di un congruo numero di esercizi durante il corso.

Risultati di apprendimento attesi

Conoscenza delle principali metodologie utili per lo studio di alcune classi di processi stocastici a tempo e spazio discreti. Capacità di utilizzare le proprietà del Processo di Poisson e i processi Markoviani in ambito modellistica. Sviluppo delle abilità necessarie per la formulazione di modelli stocastici di interesse per le applicazioni.

Frequenza

facoltativa



Testi in inglese

Tipo testo

Testo

Lingua insegnamento

Italian

Contenuti

Jointly distributed random variables.
Calculus of expectations, probabilities and variance through conditioning.
Discrete time Markov chains.
Exponential random variable and Poisson process.
Continuous time Markov chains.
Intro to Brownian motion.

Testi di riferimento

Ross S.M. Introduction to probability models. Academic Press, 2003.

Obiettivi formativi

Students will develop the necessary skills to write down simple probabilistic models of applied interest. The introduction of stochastic processes and their properties is always motivated by the wish to develop models for observed phenomena. Aim of the course include the development of the abilities for the formulation and the study of simple stochastic models

DUBLIN DESCRIPTORS (see

http://www.study-in-italy.it/php5/scheda_corso.php?ambiente=off&anno=2009&corso=1214968)

Knowledge and understanding: The course introduces first topics in Stochastic Processes Theory making use of basic knowledge learnt in the Probability and Statistics course (goal 3). The aim is to give to the student the necessary tools for the development of simple stochastic models and for their study (goal 17). Some model simulation exercises request the development of simple software programs. The choice of the software is made by the student in accordance with his computational and computer science skills (goal 18). The textbook is in English and its use is common to many universities in the world. This choice helps the student to get used to English for scientific purposes

Applying knowledge and understanding:

The introduction of stochastic processes theory is performed avoiding the use of measure theory concepts. Too abstract concepts are not considered in this course. The discussed results can be dealt at the level of an undergraduate student. This approach is not incompatible with a good understanding of important concepts and with a rigorous work on them. The student becomes used to proof by himself results analogous to those discussed during the lessons (goal 1). Main goal of the course is to help the student to develop skills necessary for stochastic problem solving. To attain this aim exercises are regularly assigned and individually corrected. A discussion in class on the most interesting mistakes helps to improve individual criticism. These exercises are considered as part of the final exam (goals 2,3,4). Sometimes the solution of exercises requests the use of computational tools or suitable software (goal 5)

Making judgements:

Teamwork is allowed for the solution of assigned exercises. Group discussions home or during the lessons are encouraged to develop logic abilities and to improve communication skills (goals 1 and 4). Often the solution technique for the proposed exercises is not unique. Students are invited to present their solution to the class. Abilities in recognizing possible mistakes and understanding details in alternative proofs are developed through exercises discussions (goal 3)

Communication

Communication skills are improved defending individual approach to the

Tipo testo

Testo

exercise solution (goal 1 for Italian language). The student improves his ability to communicate with non mathematical people through his effort to model features of physical, computer science or biological facts and reporting the results suggested by the formulated model (goal 2). The textbook is in English, making the student used to English for scientific purposes (goal 3)

Learning skills .

The concepts introduced in this course will be useful to students that will pursue their studies at the master level. Indeed they will have clear examples in their mind for the more abstract concepts that will be part of their future studies (goal 2). The same concept, with special regard to modeling skills, will be useful for many different job activities (goal 2). Learning the scientific method is basic to develop mathematical models and it will be useful in the future activities of the student, to formalize different facts (goal 4).

Modalità di verifica dell'apprendimento

Oral examination including the solution of an exercise. The exercise in the oral can be skipped if homework are done during the course

Risultati di apprendimento attesi

Knowledge of methods useful to study some classes of stochastic processes. Ability in using Poisson and Markov processes to model observed facts. Development of abilities useful to propose and study stochastic models of applied interest.

Frequenza

optional

Testi del Syllabus

Docente	PIRONTI MARCO	Matricola: 014125
Anno offerta:	2014/2015	
Insegnamento:	MFN0946 - ECONOMIA E GESTIONE DELLE IMPRESE NET BASED (Management of Net Based Enterprises)	
Corso di studio:	008515 - INFORMATICA (Computer science)	
Anno regolamento:	2014	
CFU:	6	
Settore:	SECS-P/08	
Tipo attività:	C - Affine/Integrativa	
Partizione studenti:	-	
Periodo:	Secondo Semestre	



Testi in italiano

Tipo testo	Testo
Lingua insegnamento	Italiano
Contenuti	<p>La comunicazione è una funzione chiave e la gestione della comunicazione incide sulla visione che il cliente ha del prodotto. Per questo dopo aver ideato e creato un prodotto o servizio è utile che l'innovazione venga comunicata nel modo più appropriato ai consumatori:</p> <p>syllabus</p> <p>capire e sviluppare il processo di marketing defiire la strategia e gli obiettivi di marketing integrare il marketing communication plan gestire i canali del marketing e integrarli.</p>
Testi di riferimento	Marketing plans how to prepare them how to use them seventh edition Malcomlm mcdonald and hugh Wilson
Obiettivi formativi	il corso ha l'obiettivo di conferire agli studenti gli strumenti per 1) Creare una campagna marketing 2) Comunicare il valore al proprio consumatore
Propedeutico a	Nulla
Modalità di verifica dell'apprendimento	La verifica dell'apprendimento avverrà attraverso un esame scritto in aula e una tesina sviluppata a gruppi all'interno del corso.



Testi in inglese

Tipo testo	Testo
Lingua insegnamento	Italian
Contenuti	Communication is a key function in the organization and the management of communication affects the vision that the customer has of the product and organization brand . For this reason, after having conceived and created a product or service is important to communicate and advertise it in the most appropriate way for consumers: syllabus understand and develop the marketing process define the strategy and marketing objectives integrated marketing communication plan manage and integrate marketing channels.
Testi di riferimento	Marketing plans how to prepare them how to use them seventh edition Malcom I. McDonald and Hugh Wilson
Obiettivi formativi	The course aims to give students the tools to 1) Create a Marketing Campaign 2) Communicating the value to its consumers
Propedeutico a	Nothing
Modalità di verifica dell'apprendimento	A written assessment will be required to test the learning level

Testi del Syllabus

Docente	BALOSSINO NELLO	Matricola: 010522
Anno offerta:	2014/2015	
Insegnamento:	MFN0972 - ELABORAZIONE DI IMMAGINI E VISIONE ARTIFICIALE (Image Processing and Artificial Vision)	
Corso di studio:	008515 - INFORMATICA (Computer science)	
Anno regolamento:	2014	
CFU:	9	
Settore:	INF/01	
Tipo attività:	B - Caratterizzante	
Partizione studenti:	-	
Anno corso:	1	
Periodo:	Primo Semestre	



Testi in italiano

Tipo testo

Testo

Lingua insegnamento

Italiano

Contenuti

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.

Testi di riferimento

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.

S. J. D. Prince, "Computer Vision Models, Learning, and Inference", Cambridge University Press, 2012.

W. K. Pratt, "Introduction to Digital Image Processing", CRC Press, 2013

J. C. Russ, "The Image Processing Handbook", Sixth ed., CRC Press, 2011.

Tipo testo

Testo

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.

Prerequisiti

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

Propedeutico a

Preparazione della prova finale in una visione interdisciplinare.

Note

Sono forniti dal docente slide sugli argomenti delle lezioni.

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.

Risultati di 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.

Frequenza

La frequenza è consigliata ma non è obbligatoria.



Testi in inglese

Tipo testo

Testo

Lingua insegnamento

Italian

Contenuti

Introduction Image Processing definition and field of application. Fundamental Steps in Digital Image Processing. Components of an Image Processing System.

Digital images: fundamentals. Light and the Electromagnetic Spectrum. Image Sensing and Acquisition. Spatial and Intensity Resolution. An overview of the mathematical tools used in image processing.

Intensity transformation and spatial filtering. Transformation and spatial filtering examples. Histogram processing. Histogram matching. Local Histogram Processing. Fundamentals of Spatial Filtering. Smoothing Spatial Filters. Sharpening Spatial Filters. Combining Spatial Enhancement Methods.

Filtering in the Frequency Domain. Background. The Fourier Transform of Functions of One Continuous Variable. Sampling and the Fourier Transform of Sampled Functions. Extension to Functions of Two Variables. The Discrete Fourier Transform (DFT) and its properties. The Basics of Filtering in the Frequency Domain.

Image Smoothing Using Frequency Domain Filters. Image Sharpening Using Frequency Domain Filters. The Fast Fourier Transform.

Image Restoration and Reconstruction. Periodic Noise Reduction by Frequency Domain Filtering. Linear, Position-Invariant Degradations. Inverse filtering. Image Reconstruction from Projections.

Color image processing. Color fundamentals. Pseudo-color image processing. Color transformations.

Morphological Image Processing. Erosion and dilation. Some simple morphological algorithms.

Image Segmentation. Point, Line, and Edge Detection. The Canny edge detector. Thresholding. Region-Based Segmentation. Region Splitting and Merging.

Representation and Description. Chain codes. Signature. Skeleton. Edge descriptors. Region descriptors. Principal components usage in description.

Robotic Vision. Elements of visual perception. Stereoscopic vision.

Testi di riferimento

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.

S. J. D. Prince, "Computer Vision Models, Learning, and Inference", Cambridge University Press, 2012.

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

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

Tipo testo

Testo

Obiettivi formativi

Course topics aim to give the student mathematical and computer science tools useful for image processing in the spatial and frequency domains. The image processing tasks target image enhancement and restoration, object recognition and artificial vision systems. The course presents also an experimental part using the MATLAB programming environment and the EidosLab software.

Prerequisiti

Lessons assume that the students has knowledge of vector operations and calculus, matrix operations, and mathematical analysis. The experimental part requires basic programming skills.

Propedeutico a

Preparation for the final exam with an interdisciplinary attitude.

Note

Apart from the suggested textbooks, slides from the teacher covering the lessons are available.

Modalità di verifica dell'apprendimento

The student is required to prepare a project and implement it to show his/her abilities in performing image processing operations to reach a goal; moreover an oral exam will touch all the topics of the course.

Risultati di apprendimento attesi

The student will learn to use algorithms for image enhancement in the spatial and transformed domains, and for object description and recognition. Moreover the student will be able to analyze and design artificial vision systems.

Frequenza

Attendance to the lessons is suggested but not mandatory.

Testi del Syllabus

Docente	LOMBARDO VINCENZO	Matricola: 013498
Anno offerta:	2014/2015	
Insegnamento:	MFN0974 - ELABORAZIONE DIGITALE AUDIO E MUSICA (Digital sound and music processing)	
Corso di studio:	008515 - INFORMATICA (Computer science)	
Anno regolamento:	2014	
CFU:	6	
Settore:	INF/01	
Tipo attività:	B - Caratterizzante	
Partizione studenti:	-	
Anno corso:	1	
Periodo:	Secondo Semestre	

Tipo testo	Testo
Lingua insegnamento	Italiano
Contenuti	<p>Il corso consta di due moduli.</p> <p>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.</p> <p>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.</p>
Testi di riferimento	<p>Prima unità: V. Lombardo, A. Valle, Audio e multimedia, IV edizione, Apogeo Education, Maggioli Editore, Milano, 2014 (Capp. 1, 2, 3, 4, 5, 6, 8).</p> <p>Seconda unità: V. Lombardo, A. Valle (chap. 5 e 9). A. Valle, SCIRMA: the SuperCollider Italian Manual at CIRMA, disponibile presso http://www.cirma.unito.it/andrea/sw/sc.html</p>
Obiettivi formativi	<p>Il corso si propone di fornire agli studenti un approfondimento sui seguenti argomenti:</p> <p>Concetti fondamentali dell'acquisizione, rappresentazione, elaborazione, compressione del suono mediante strumenti informatici.</p> <ul style="list-style-type: none">• 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.
Prerequisiti	Corsi base di informatica.
Propedeutico a	Nessuno
Note	Tenuto presso il DAMS (Dipartimento Studi Umanistici)
Modalità di verifica dell'apprendimento	Orale, prova di laboratorio
Risultati di apprendimento attesi	<p>Competenze in</p> <ul style="list-style-type: none">• acquisizione, rappresentazione, elaborazione, compressione del suono mediante strumenti informatici;• protocollo MIDI;• tecniche di sintesi digitale del segnale audio e di composizione algoritmica;• linguaggi di programmazione e editor per l'elaborazione digitale del segnale audio.

Tipo testo**Testo****Frequenza**

Opzionale

Testi in inglese

Tipo testo	Testo
Lingua insegnamento	Italian
Contenuti	<p>The course consists of two units.</p> <p>The first unit is an introduction to the fundamental notions of acquisition, representation, processing, and compression of sound through digital tools. Moreover, the unit deals with the MIDI protocol for the representation of music at a simboli level. Teaching consists in theoretical lectures and laboratory practice.</p> <p>The second unit provides an introduction to the techniques of digital synthesis of the audio signal as well as the algorithmic composition, in a wide perspective that includes music, sound design and multimedia. With these goals the unit introduces the programmino language SuperCollider, that currently is the most powerful and versatile tool for the automatic control of sound materials generation and organization.</p>
Testi di riferimento	<p>First unit: V. Lombardo, A. Valle, Audio e multimedia, IV edizione, Apogeo Education, Maggioli First unit: V. Lombardo, A. Valle, Audio e multimedia, IV edizione, Apogeo Education, Maggioli Editore, Milano, 2014 (Capp. 1, 2, 3, 4, 5, 6, 8).</p> <p>Second unit: V. Lombardo, A. Valle, Audio e multimedia, IV edizione, Apogeo Education, Maggioli Editore, Milano, 2008 (chap. 5 e 9). A. Valle, SCIRMA: the SuperCollider Italian Manual at CIRMA, available at http://www.cirma.unito.it/andrea/sw/sc.html</p>
Obiettivi formativi	<p>The course provides students with deep knowledge on the following topics:</p> <ul style="list-style-type: none">• Fundamental knowledge about acquisition, representation, processing and compression of sound through computer tools;• MIDI protocol for the representation of music at the symbolic level;• Techniques of digital synthesis of the sudio signal and of algorithmic composition, in a practical wide perspective, that includes the musical area and the application to sound design and multimedia;• Programming languages and editors for the digital processing of the audio signal.
Prerequisiti	Basic courses of informatics.
Propedeutico a	None
Note	To be attended at DAMS structure (Department of Humanities)
Modalità di verifica dell'apprendimento	Oral, laboratory project
Risultati di apprendimento attesi	<p>Competences in</p> <ul style="list-style-type: none">• acquisition, representation, processing and compression of sound through computer tools;• MIDI protocol;• Techniques of digital synthesis of the sudio signal and of algorithmic composition;• Programming languages and editors for the digital processing of the audio signal.

Tipo testo**Testo****Frequenza**

Optional

Testi del Syllabus

Docente	MAGGIORA MARCO	Matricola: 061389
Anno offerta:	2014/2015	
Insegnamento:	MFN0970 - FISICA PER LE APPLICAZIONI DI REALTA' VIRTUALE (Physics for Virtual Reality Applications)	
Corso di studio:	008515 - INFORMATICA (Computer science)	
Anno regolamento:	2014	
CFU:	6	
Settore:	FIS/01	
Tipo attività:	C - Affine/Integrativa	
Partizione studenti:	-	
Anno corso:	1	
Periodo:	Primo Semestre	



Testi in italiano

Tipo testo

Testo

Lingua insegnamento

Italiano

Contenuti

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 1.5 Moto circolare, uniforme e non). 1.6 Moto relativo

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 vincolare. 2.6 Piano inclinato.

3. Forze d'attrito
3.1 Attrito statico e dinamico. 3.2 Equilibrio di un corpo su un piano inclinato. 3.3 Moto circolare con attrito

4. Lavoro ed energia
Conservazione dell'energia meccanica. 4.1 Lavoro di forze costanti e variabili. 4.2 Energia cinetica. 4.3 Forze conservative, non conservative, e dissipative. 4.4 Energia potenziale. (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. 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

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.

Tipo testo

Testo

13. Meccanica ondulatoria
13.1 Natura e propagazione delle onde. 13.2 Onde meccaniche.
. 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. 14.9 Effetti supersonici. 14.10 Effetto doppler. 14.11 Applicazioni ultrasoniche militari. 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. 15.6 Effetti di rifrazione. 15.7 Effetti di riflessione totale. 15.8 Lenti sferiche sottili.: 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. 16.3 Fenomeni di dispersione: prisma e arcobaleno. 16.4 Ottica dei miraggi. 16.5 Interferenza su lamine sottili.

17. Applicazioni ottiche tecnologiche e medicali
17.1 La macchina fotografica e gli obiettivi fotografici. 17.2 L'occhio umano. 17.3 Applicazioni oculistiche. 17.4 Aberrazione di lenti e specchi. 17.5 Strumenti ottici complessi.. 17.6 Limite di risoluzione di strumenti ottici.

Testi di riferimento

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

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.

Prerequisiti

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

Propedeutico a

nessuno

Modalità di verifica dell'apprendimento

Esame orale

Risultati di 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 e nell'introduzione di effetti più realistici in videogiochi o ambientazioni di realtà virtuale.

Frequenza

Facoltativa



Testi in inglese

Tipo testo

Testo

Lingua insegnamento

Italian

Contenuti

1. Kinematics
1.1 Space, speed, acceleration 1.2 Vectors 1.3 Motion along a straight line: uniform velocity, uniform speed 1.4 Parabolic motion 1.5 Circular motion, uniform and not 1.6 Relative motion

2. Fundamental dynamics
2.1 Law of Dynamics 2.2 Gravitation: Newton's law, Cavendish' experience 2.3 Geosynchronous satellite: speed, orbit 2.4 Inertial and not-inertial reference frames 2.5 Constrains and forces 2.6 Inclined plane

3. Friction
3.1 Static and dynamic friction. 3.2 Equilibrium on an inclined rough plane 3.3 Circular motion and friction.

4. Energy and work
4.1 Work 4.2 Kinetic energy 4.3 Conservative and dissipative forces 4.4 Potential energy 4.5 Conservation of energy 4.6 Motion description through energy conservation 4.7 Energy and friction

5. Quantity of motion
5.1 Quantity of motion (conservation) 5.2 Impulse and forces; impulse theorem 5.3 Fixed and variable motion; the rocket

6. Elastic and inelastic scattering
6.1 Scattering and energy 6.2 Ballistic pendulum 6.3 Bi-dimensional scattering

7. Centre of mass
7.1 Translation, law of dynamics. 7.2 Centre of gravity 7.3 Human body centre of mass.

8. Translational and rotational motion
8.1 Torque 8.2 Moment of inertia 8.3 Rotational dynamics 8.4 Motion of the pulley 8.5 Constraints and torque 8.6 Rotational kinetic energy 8.7 Translational and rotational motion on an inclined plane 8.8 Roto-translation and friction 8.9 Angular momentum conservation 8.10 Applications to human body

9. Statics and equilibrium
9.1 Applications to muscles and joints 9.2 Applications to architecture

10. Elasticity, strain and deformation

11. Elasti, shear and compression modulus
11.1 Equilibrium in traction and compression 11.2 Applications to muscles and joints 11.3 Applications to architecture

12. Harmonic motion
12.1 Harmonic oscillator: simple and forced. 12.2 Energetic description of the harmonic oscillator.

13. Wave mechanics
13.1 Waves and propagation 13.2 Mechanical waves: transverse, longitudinal and surface waves 13.3 Energy, power and intensity 13.4 Reflection and transmission

14. Acoustics
14.1 Sound 14.2 Intensity 14.3 The human ear 14.4 Sensibility 14.5

Tipo testo

Testo

Interference 14.6 Stationary waves 14.7 Batiments 14.8 Musical instruments: string and winds; timbric analysis 14.9 Supersonic effects 14.10 Doppler effect 14.11 Military applications: sonar 14.12 Medical applications: scan, eco-doppler

15. Geometrical Optics

15.1 Rays model 15.2 Refraction and reflection; Snell's law 15.3 Virtual and real images 15.4 Planar mirrors 15.5 Spherical mirrors; mirrors law 15.6 Refractive effects 15.7 Total reflection: prisms and optic fibers 15.8 Thin spheric lenses; lenses law 15.9 Systems of thin lenses 15.10 Lenses constructor law

16. Wave optics

16.1 Huygens' principle; Snell's law deduction 16.2 Diffraction and interference: Young's experience; diffraction on objects; reticula 16.3 Dispersion: prisms and rainbow 16.4 Optics of mirages 16.5 Interference on thins foil: soap bubbles, Newton's ring, antiglare treatment

17. Optical applications

17.1 Camera and lenses 17.2 Human eye: anatomy, ametropies 17.3 Ophthalmic applications 17.4 Aberrations of lenses and mirrors 17.5 Complex optical systems: hand lens, microscope, telescopes 17.6 Resolution of optical instruments

Testi di riferimento

1. GIANCOLI Fisica Casa Editrice Ambrosiana

or any other text of Physics for Science and/or Engineering as for example:

2. SERWAY Fisica 1 EdiSes

3. RESNICK, HALLIDAY, KRANE Fisica 1 Casa Editrice Ambrosiana

Obiettivi formativi

The student will get access to simple applications of Physics that can be understood within the framework of a basic Physics knowledge, from mechanics to optics, learning how to spot the underlying Physics principles and laws and to develop the mathematical equations needed to describe such applications.

Prerequisiti

Basic Mathematics: equations, trigonometry, vector calculus, fundamental derivatives and integrals.

Propedeutico a

none

Modalità di verifica dell'apprendimento

Oral examination

Risultati di apprendimento attesi

A good knowledge of those physical phenomena occurring in all day life or that belongs to the great technological discoveries of the XX century, in order to have a creative approach in producing didactical tools for Physics, or in dressing videogames and virtual reality environments with more realistic effects.

Frequenza

Optional

Testi del Syllabus

Docente	TORASSO PIETRO	Matricola: 010688
Anno offerta:	2014/2015	
Insegnamento:	MFN0942 - INTELLIGENZA ARTIFICIALE E LABORATORIO (Artificial Intelligence (with experimentations))	
Corso di studio:	008515 - INFORMATICA (Computer science)	
Anno regolamento:	2014	
CFU:	9	
Settore:	INF/01	
Tipo attività:	B - Caratterizzante	
Partizione studenti:	-	
Periodo:	Secondo Semestre	



Testi in italiano

Tipo testo

Testo

Contenuti

- Ragionamento Automatico (meccanismi di ragionamento per calcolo proposizionale, per calcolo dei predicati del primo ordine, logic programming, ragionamento con conoscenza incompleta)

- Laboratorio su uso di formalismi logici per la risoluzione automatica di problemi e rappresentazione della conoscenza (con particolare attenzione a Prolog e Answer Set Programming)

- Pianificazione automatica (rappresentazione delle azioni e degli obiettivi, metodi per la generazione automatica di piani, pianificazione e azione, piani condizionali, ripianificazione)

- Ragionamento in presenza di incertezza (ragionamento probabilistico, reti bayesiane e metodi di ragionamento su reti bayesiane, esempi di uso di ambienti software basati su reti bayesiane, ragionamento probabilistico in sistemi che evolvono nel tempo).

- Laboratorio per lo sviluppo di un agente intelligente in grado di esibire sia comportamenti deliberativi che reattivi in un ambiente parzialmente osservabile utilizzando ambienti basati su sistemi a regole di produzione.

Il programma dettagliato sarà definito durante il corso.

Testi di riferimento

L. Console, E. Lamma, P. Mello, M. Milano: "Programmazione logica e Prolog" (nuova edizione), UTET 1997

S. J. Russell e P. Norvig, Artificial Intelligence, Second edition, 2003, Prentice Hall

Obiettivi formativi

Il corso ha l'obiettivo di approfondire le conoscenze di Intelligenza Artificiale con particolare riguardo a meccanismi di ragionamento e di rappresentazione della conoscenza. Alle competenze metodologiche si affiancano competenze progettuali perche' 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).



Testi in inglese

Tipo testo

Testo

Contenuti

The course covers the following topics:

- Automated Reasoning (methods for propositional calculus, First order predicate calculus, logic programming, reasoning with incomplete knowledge)
- Experimentation of logic formalisms for automated problem solving and knowledge representation and reasoning (particular attention will be devote to use the functionalities offered by PROLOG for developing software demos)
- Automated planning (representation of actions and goals, methods for automatic plan generation, planning and acting, conditional plans, replanning)
- Reasoning under uncertainty (probabilistic reasoning, Bayesian networks and reasoning mechanisms based on Bayesian networks, exercises of software environments exploiting Bayesian networks, probabilistic reasoning over time)
- Experimentation for the development of a proof of concept of an intelligent agent able to exhibit deliberative and reactive behavior in a partially observable environment. The proof of concept will be developed by exploiting facilities offered by software tools based on production rules.

Testi del Syllabus

Docente	TORASSO PIETRO	Matricola: 010688
Anno offerta:	2014/2015	
Insegnamento:	MFN0942 - INTELLIGENZA ARTIFICIALE E LABORATORIO (Artificial Intelligence (with experimentations))	
Corso di studio:	008515 - INFORMATICA (Computer science)	
Anno regolamento:	2014	
CFU:	9	
Settore:	INF/01	
Tipo attività:	B - Caratterizzante	
Partizione studenti:	-	
Anno corso:	1	
Periodo:	Secondo Semestre	



Testi in italiano

Tipo testo	Testo
Lingua insegnamento	Italiano
Contenuti	<p>Il corso copre le seguenti problematiche</p> <ul style="list-style-type: none">- Ragionamento Automatico (meccanismi di ragionamento per calcolo proposizionale, per calcolo dei predicati del primo ordine, logic programming, ragionamento con conoscenza incompleta)- Laboratorio su uso di formalismi logici per la risoluzione automatica di problemi e rappresentazione della conoscenza (con particolare attenzione a Prolog e Answer Set Programming)- Pianificazione automatica (rappresentazione delle azioni e degli obiettivi, metodi per la generazione automatica di piani, pianificazione in presenza di conoscenza incompleta e/o osservabilità parziale, piani condizionali, ripianificazione)- Ragionamento in presenza di incertezza (ragionamento probabilistico, reti bayesiane e metodi di ragionamento su reti bayesiane, esempi di uso di ambienti software basati su reti bayesiane).- Laboratorio per lo sviluppo di un agente intelligente in grado di esibire sia comportamenti deliberativi che reattivi in un ambiente parzialmente osservabile utilizzando ambienti basati su sistemi a regole di produzione. <p>Il programma dettagliato sarà definito durante il corso.</p>
Testi di riferimento	<p>S. J. Russell e P. Norvig: "Artificial Intelligence: A modern approach", Third edition, Prentice Hall</p> <p>L. Console, E. Lamma, P. Mello, M. Milano: "Programmazione logica e Prolog" (nuova edizione), UTET 1997</p>
Obiettivi formativi	<p>Il corso ha l'obiettivo di approfondire le conoscenze di Intelligenza Artificiale con particolare riguardo a meccanismi di ragionamento e di pianificazione automatica. 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 programmazione a regole).</p>
Prerequisiti	<p>E' utile avere seguito un corso di introduzione all'Intelligenza Artificiale (tipo il corso di Sistemi Intelligenti offerto nella laurea triennale a UniTo).</p> <p>La conoscenza di algoritmi di ricerca su grafi e delle nozioni elementari su formalismi logici.</p> <p>E' richiesta esperienza di programmazione e di sviluppo di (piccoli) sistemi prototipali</p>
Propedeutico a	Il corso fornisce solide basi per il corso di Agenti Intelligenti
Modalità di verifica dell'apprendimento	Esame orale che comprende sia la parte metodologica del corso che la presentazione e la discussione critica dei progetti ed esercizi svolti in laboratorio
Frequenza	Non obbligatoria, ma fortemente consigliata



Testi in inglese

Tipo testo	Testo
Lingua insegnamento	Italian
Contenuti	<p>The course covers the following topics:</p> <ul style="list-style-type: none">• Automated Reasoning (methods for propositional calculus, First order predicate calculus, logic programming, reasoning with incomplete knowledge)• Experimentation of logic formalisms for automated problem solving and knowledge representation and reasoning (particular attention will be devoted to the functionalities offered by PROLOG and Answer Set programming for developing software demos)• Automated planning (representation of actions and goals, methods for automatic plan generation, planning and acting, conditional plans, replanning)• Reasoning under uncertainty (probabilistic reasoning, Bayesian networks and reasoning mechanisms based on Bayesian networks, exercises of software environments exploiting Bayesian networks)• Experimentation for the development of a proof of concept of an intelligent agent able to exhibit deliberative and reactive behavior in a partially observable environment. The proof of concept will be developed by exploiting facilities offered by software tools based on production rules.
Testi di riferimento	<p>S. J. Russell e P. Norvig: "Artificial Intelligence: A modern approach", Third edition, Prentice Hall</p> <p>L. Console, E. Lamma, P. Mello, M. Milano: "Programmazione logica e Prolog" (nuova edizione), UTET 1997</p>
Obiettivi formativi	<p>The course is an advanced course on Artificial Intelligence and it aims at covering advanced topics concerning reasoning mechanisms and automated planning.</p> <p>The course will supplement the methodological aspects with an experimental part covering the use of logic programming in problem solving and reasoning and a project for developing a proof of concept for an intelligent agent.</p>
Prerequisiti	<p>An introductory course on Artificial Intelligence (covering automated problem solving and elements of knowledge representation and reasoning) is quite useful.</p> <p>Experience in programming and developing proof of concepts is desirable.</p>
Propedeutico a	The course provides a good basis for the course on "Intelligent Agents"
Modalità di verifica dell'apprendimento	Oral examination covering both the methodological issues presented in the course and the experimental part. In particular the exercises and the project developed for the course will be discussed
Frequenza	Not compulsory, but strongly suggested

Testi del Syllabus

Docente	GROSSO ANDREA CESARE	Matricola: 014296
Anno offerta:	2014/2015	
Insegnamento:	MFN1473 - ISTITUZIONI DI CALCOLO MATRICIALE E RICERCA OPERATIVA (Additional Matrix Calculus and Operational Research)	
Corso di studio:	008515 - INFORMATICA (Computer science)	
Anno regolamento:	2014	
CFU:	6	
Settore:	MAT/09	
Tipo attività:	D - A scelta dello studente	
Partizione studenti:	-	
Periodo:	Primo Semestre	



Testi in italiano

Tipo testo	Testo
Lingua insegnamento	Italiano
Contenuti	Vettori e matrici. Operazioni fondamentali. Cenni di geometria in spazi vettoriali. Combinazioni lineari, indipendenza lineare. Sottospazi e basi. Insiemi convessi, poliedri. Programmazione lineare. Modellazione. Struttura della regione ammissibile. Soluzioni di base. Algoritmo del simplesso. Dualità. Cenni di programmazione a variabili intere.
Testi di riferimento	Appunti forniti dal docente.
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 scientifico fondamentale. La ricerca operativa studia modelli e metodi per l'utilizzo ottimale di risorse scarse (in ambiti produttivi, finanziari, ecc.).
Prerequisiti	--
Propedeutico a	--
Modalità di verifica dell'apprendimento	Esame scritto e orale.
Risultati di apprendimento attesi	Padronanza delle tecniche algebriche di base e degli algoritmi per programmazione lineare a variabili continue. Capacità di modellare in programmazione lineare.



Testi in inglese

Tipo testo	Testo
Lingua insegnamento	Italian
Contenuti	Vectors and matrices. Basic operations. Basics of geometry in vector spaces. Linear combinations, linear independence. Subspaces, bases of vector (sub)spaces. Convex sets and polyhedra. Linear programming. Modeling. Structure of the feasible region. Basic feasible solutions. Duality. The simplex algorithm for linear programs. Basics of integer linear programming.
Testi di riferimento	Lecture notes provided by the instructor.
Obiettivi formativi	Matrix calculus and linear algebra constitute a fundamental tool in any scientific field. Operational research deals mainly with rational use of scarce resources in real world problems. The course teaches basics in both such areas.
Prerequisiti	..
Propedeutico a	--
Modalità di verifica dell'apprendimento	Written and oral exams.
Risultati di apprendimento attesi	The student should be able to master the basic techniques in linear algebra and linear programming. He/she should also acquire the skill of properly modeling real-world problems producing the appropriate linear programs.

Testi del Syllabus

Docente	ZACCHI MADDALENA	Matricola: 011333
Anno offerta:	2014/2015	
Insegnamento:	MFN0985 - ISTITUZIONI DI LINGUAGGI FORMALI (Additional Formal Languages)	
Corso di studio:	008515 - INFORMATICA (Computer science)	
Anno regolamento:	2013	
CFU:	6	
Settore:	INF/01	
Tipo attività:	B - Caratterizzante	
Partizione studenti:	-	
Periodo:	Primo Semestre	



Testi in italiano

Tipo testo	Testo
Lingua insegnamento	Italiano
Contenuti	<p>Concetti centrali della teoria dei linguaggi formali: alfabeto, stringa, linguaggio.</p> <p>Automi a stati finiti: descrizione informale, automi a stati finiti deterministici, non deterministici e con epsilon transizioni. Equivalenza dei diversi tipi di automi a stati finiti e loro applicazioni nelle ricerche testuali.</p> <p>Espressioni e linguaggi regolari. Applicazioni delle espressioni regolari e loro proprietà algebriche.</p> <p>Proprietà dei linguaggi regolari: proprietà di chiusura e problemi di decisione.</p> <p>Automi a stati finiti riconoscitori dei linguaggi regolari, minimizzazione di automi.</p> <p>Grammatiche e linguaggi liberi dal contesto: alberi sintattici, ambiguità nelle grammatiche e nei linguaggi. Automi a pila non deterministici e deterministici.</p> <p>Automi a pila riconoscitori dei linguaggi liberi dal contesto. Proprietà dei linguaggi liberi.</p> <p>Struttura di un compilatore</p> <p>Analisi sintattica top-down: parsing predittivo e grammatiche LL(1), parsing a discesa ricorsiva.</p> <p>Analisi sintattica bottom-up: parsing shift-reduce, parsificazione LR semplice.</p> <p>Traduzione guidata dalla sintassi: attributi ereditati e sintetizzati, ordine di valutazione, schemi di traduzione. Implementazione di definizioni guidate dalla sintassi L-attribuite, traduzione durante il parsing a discesa ricorsiva.</p> <p>Generazione del codice intermedio: codice a tre indirizzi. Traduzione di espressioni aritmetiche, espressioni booleane e traduzione di istruzioni.</p>
Testi di riferimento	<p>J. E. Hopcroft, R. Motwani, J. D. Ullman, "Automi, Linguaggi e Calcolabilità", Pearson Paravia Bruno Mondadori S.p.A., 2009, ISBN 978-88-7192-552-3.</p> <p>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.</p> <p>Terence Parr, The Definitive ANTLR Reference: Building Domain-Specific Languages, The Pragmatic Bookshelf, 2007, ISBN 978-0978739256.</p>
Obiettivi formativi	<p>Il corso di "Linguaggi Formali e Traduttori" si propone 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.</p> <p>Le metodologie e le tecniche presentate nel corso 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.</p>
Prerequisiti	<p>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.</p>
Propedeutico a	<p>Il corso introduce nozioni fondamentali utili in molti contesti ed essenziali per la formazione di un laureato in Informatica.</p>
Modalità di verifica dell'apprendimento	<p>L'esame consiste di una prova scritta comprendente una prima parte di domande di teoria e una seconda di esercizi da risolvere, analoghi a quelli assegnati e corretti durante le esercitazioni.</p>

Tipo testo**Risultati di apprendimento attesi*****Testo***

Conoscenza dei principi fondamentali della sintassi e della semantica dei linguaggi di programmazione e delle principali metodologie di parsificazione e traduzione. Conoscenza di strumenti per la progettazione di compilatori di linguaggi di programmazione.



Testi in inglese

Tipo testo

Testo

Lingua insegnamento

Italian

Contenuti

Central concepts of formal languages: alphabet, string, language.
Finite state automata: informal description, deterministic, non-deterministic and with epsilon transitions finite automata. Equivalence of different types of finite state automata and their applications in text searches.
Regular expressions and languages. Applications of regular expressions and their algebraic properties. Properties of regular languages: closure properties and decision problems.
Finite state automata recognizers of regular languages, minimization of automata.
Grammars and context-free languages: parse trees, ambiguities in grammars and languages.
Deterministic and non-deterministic pushdown automata. Pushdown automata recognizers of context-free languages. Properties of context-free languages.
Structure of a compiler.
Top-down parsing: predictive parsing and LL(1) grammars, recursive descent parsing.
Bottom-up parsing: shift-reduce parsing, simple LR parsing.
Syntax-directed translation: inherited and synthesized attributes, evaluation order, syntax-directed translation schemes. Implementation of L-attributed syntax-directed definitions: translation during the recursive descent parsing.
Intermediate code generation: three-address code. Translation of arithmetic expressions, Boolean expressions and translation of statements.

Testi di riferimento

J.E. Hopcroft, R. Motwani and J.D. Ullman, "Introduction to Automata Theory, Languages and Computations", 3rd Edition, Pearson Education, Addison Wesley, 2007.
A.V. Aho, M.S. Lam, R.Sethi, J.D. Ullman. "Compilers: Principles, Techniques and Tools". 2nd Edition, Pearson Education, Addison Wesley, 2007.
Terence Parr, The Definitive ANTLR Reference: Building Domain-Specific Languages, The Pragmatic Bookshelf, 2007, ISBN 978-0978739256.

Obiettivi formativi

The course "Formal Languages and Translators" aims to provide students with an introductory overview of issues related to the definition and translation of programming languages, with particular reference to the design and construction of compilers.
The methodologies and techniques presented in the course are generally useful as general formalism to define the behavior of a system or to implement more simple translators of a compiler itself.

Prerequisiti

The student must be familiar with the basic concepts of set theory and the design of iterative and recursive algorithms. The student must also have acquired the ability to program in high-level languages.

Propedeutico a

The course introduces fundamental notions that are useful in many contexts, and are an essential background for a graduate in Computer Science.

Modalità di verifica dell'apprendimento

The examination consists of a written test, the first part of which comprises questions on theory, and the second part of which comprises exercises similar to those presented in the lectures and teaching materials.

Tipo testo***Testo*****Risultati di
apprendimento attesi**

Knowledge of the basic principles of the syntax of programming languages and the main methods of parsing and translating. Knowledge of tools for the design of compilers for programming languages.

Testi del Syllabus

Docente	ZACCHI MADDALENA	Matricola: 011333
Anno offerta:	2014/2015	
Insegnamento:	MFN0985 - ISTITUZIONI DI LINGUAGGI FORMALI (Additional Formal Languages)	
Corso di studio:	008515 - INFORMATICA (Computer science)	
Anno regolamento:	2014	
CFU:	6	
Settore:	INF/01	
Tipo attività:	B - Caratterizzante	
Partizione studenti:	-	
Periodo:	Primo Semestre	



Testi in italiano

Tipo testo	Testo
Lingua insegnamento	Italiano
Contenuti	<p>Concetti centrali della teoria dei linguaggi formali: alfabeto, stringa, linguaggio.</p> <p>Automi a stati finiti: descrizione informale, automi a stati finiti deterministici, non deterministici e con epsilon transizioni. Equivalenza dei diversi tipi di automi a stati finiti e loro applicazioni nelle ricerche testuali.</p> <p>Espressioni e linguaggi regolari. Applicazioni delle espressioni regolari e loro proprietà algebriche.</p> <p>Proprietà dei linguaggi regolari: proprietà di chiusura e problemi di decisione.</p> <p>Automi a stati finiti riconoscitori dei linguaggi regolari, minimizzazione di automi.</p> <p>Grammatiche e linguaggi liberi dal contesto: alberi sintattici, ambiguità nelle grammatiche e nei linguaggi. Automi a pila non deterministici e deterministici.</p> <p>Automi a pila riconoscitori dei linguaggi liberi dal contesto. Proprietà dei linguaggi liberi.</p> <p>Struttura di un compilatore</p> <p>Analisi sintattica top-down: parsing predittivo e grammatiche LL(1), parsing a discesa ricorsiva.</p> <p>Analisi sintattica bottom-up: parsing shift-reduce, parsificazione LR semplice.</p> <p>Traduzione guidata dalla sintassi: attributi ereditati e sintetizzati, ordine di valutazione, schemi di traduzione. Implementazione di definizioni guidate dalla sintassi L-attribuite, traduzione durante il parsing a discesa ricorsiva.</p> <p>Generazione del codice intermedio: codice a tre indirizzi. Traduzione di espressioni aritmetiche, espressioni booleane e traduzione di istruzioni.</p>
Testi di riferimento	<p>J. E. Hopcroft, R. Motwani, J. D. Ullman, "Automi, Linguaggi e Calcolabilità", Pearson Paravia Bruno Mondadori S.p.A., 2009, ISBN 978-88-7192-552-3.</p> <p>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.</p> <p>Terence Parr, The Definitive ANTLR Reference: Building Domain-Specific Languages, The Pragmatic Bookshelf, 2007, ISBN 978-0978739256.</p>
Obiettivi formativi	<p>Il corso di "Linguaggi Formali e Traduttori" si propone 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.</p> <p>Le metodologie e le tecniche presentate nel corso 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.</p>
Prerequisiti	<p>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.</p>
Propedeutico a	<p>Il corso introduce nozioni fondamentali utili in molti contesti ed essenziali per la formazione di un laureato in Informatica.</p>
Modalità di verifica dell'apprendimento	<p>L'esame consiste di una prova scritta comprendente una prima parte di domande di teoria e una seconda di esercizi da risolvere, analoghi a quelli assegnati e corretti durante le esercitazioni.</p>

Tipo testo**Risultati di apprendimento attesi*****Testo***

Conoscenza dei principi fondamentali della sintassi e della semantica dei linguaggi di programmazione e delle principali metodologie di parsificazione e traduzione. Conoscenza di strumenti per la progettazione di compilatori di linguaggi di programmazione.



Testi in inglese

Tipo testo

Testo

Lingua insegnamento

Italian

Contenuti

Central concepts of formal languages: alphabet, string, language.
Finite state automata: informal description, deterministic, non-deterministic and with epsilon transitions finite automata. Equivalence of different types of finite state automata and their applications in text searches.
Regular expressions and languages. Applications of regular expressions and their algebraic properties. Properties of regular languages: closure properties and decision problems.
Finite state automata recognizers of regular languages, minimization of automata.
Grammars and context-free languages: parse trees, ambiguities in grammars and languages.
Deterministic and non-deterministic pushdown automata. Pushdown automata recognizers of context-free languages. Properties of context-free languages.
Structure of a compiler.
Top-down parsing: predictive parsing and LL(1) grammars, recursive descent parsing.
Bottom-up parsing: shift-reduce parsing, simple LR parsing.
Syntax-directed translation: inherited and synthesized attributes, evaluation order, syntax-directed translation schemes. Implementation of L-attributed syntax-directed definitions: translation during the recursive descent parsing.
Intermediate code generation: three-address code. Translation of arithmetic expressions, Boolean expressions and translation of statements.

Testi di riferimento

.E. Hopcroft, R. Motwani and J.D. Ullman, "Introduction to Automata Theory, Languages and Computations", 3rd Edition, Pearson Education, Addison Wesley, 2007.
A.V. Aho, M.S. Lam, R.Sethi, J.D. Ullman. "Compilers: Principles, Techniques and Tools". 2nd Edition, Pearson Education, Addison Wesley, 2007.
Terence Parr, The Definitive ANTLR Reference: Building Domain-Specific Languages, The Pragmatic Bookshelf, 2007, ISBN 978-0978739256.

Obiettivi formativi

The course "Formal Languages and Translators" aims to provide students with an introductory overview of issues related to the definition and translation of programming languages, with particular reference to the design and construction of compilers.
The methodologies and techniques presented in the course are generally useful as general formalism to define the behavior of a system or to implement more simple translators of a compiler itself.

Prerequisiti

The student must be familiar with the basic concepts of set theory and the design of iterative and recursive algorithms. The student must also have acquired the ability to program in high-level languages.

Propedeutico a

The course introduces fundamental notions that are useful in many contexts, and are an essential background for a graduate in Computer Science.

Modalità di verifica dell'apprendimento

The examination consists of a written test, the first part of which comprises questions on theory, and the second part of which comprises exercises similar to those presented in the lectures and teaching materials.

Tipo testo***Testo*****Risultati di
apprendimento attesi**

Knowledge of the basic principles of the syntax of programming languages and the main methods of parsing and translating. Knowledge of tools for the design of compilers for programming languages.

Testi del Syllabus

Docente	ZACCHI MADDALENA	Matricola: 011333
Anno offerta:	2014/2015	
Insegnamento:	MFN0985 - ISTITUZIONI DI LINGUAGGI FORMALI (Additional Formal Languages)	
Corso di studio:	008515 - INFORMATICA (Computer science)	
Anno regolamento:	2014	
CFU:	6	
Settore:	INF/01	
Tipo attività:	B - Caratterizzante	
Partizione studenti:	-	
Anno corso:	1	
Periodo:	Primo Semestre	



Testi in italiano

Tipo testo

Testo

Lingua insegnamento

Italiano

Contenuti

Concetti centrali della teoria dei linguaggi formali: alfabeto, stringa, linguaggio.

Automi a stati finiti: descrizione informale, automi a stati finiti deterministici, non deterministici e con epsilon transizioni. Equivalenza dei diversi tipi di automi a stati finiti e loro applicazioni nelle ricerche testuali.

Espressioni e linguaggi regolari. Applicazioni delle espressioni regolari e loro proprietà algebriche.

Proprietà dei linguaggi regolari: proprietà di chiusura e problemi di decisione.

Automi a stati finiti riconoscitori dei linguaggi regolari, minimizzazione di automi.

Grammatiche e linguaggi liberi dal contesto: alberi sintattici, ambiguità nelle grammatiche e nei linguaggi. Automi a pila non deterministici e deterministici.

Automi a pila riconoscitori dei linguaggi liberi dal contesto. Proprietà dei linguaggi liberi.

Struttura di un compilatore

Analisi sintattica top-down: parsing predittivo e grammatiche LL(1), parsing a discesa ricorsiva.

Analisi sintattica bottom-up: parsing shift-reduce, parsificazione LR semplice.

Traduzione guidata dalla sintassi: attributi ereditati e sintetizzati, ordine di valutazione, schemi di traduzione. Implementazione di definizioni guidate dalla sintassi L-attribuite, traduzione durante il parsing a discesa ricorsiva.

Generazione del codice intermedio: codice a tre indirizzi. Traduzione di espressioni aritmetiche, espressioni booleane e traduzione di istruzioni.

Testi di riferimento

J. E. Hopcroft, R. Motwani, J. D. Ullman, "Automi, Linguaggi e Calcolabilità", Pearson Paravia Bruno Mondadori S.p.A., 2009, ISBN 978-88-7192-552-3.

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.

Terence Parr, The Definitive ANTLR Reference: Building Domain-Specific Languages, The Pragmatic Bookshelf, 2007, ISBN 978-0978739256.

Obiettivi formativi

Il corso di "Linguaggi Formali e Traduttori" si propone 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 nel corso 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.

Prerequisiti

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.

Propedeutico a

Il corso introduce nozioni fondamentali utili in molti contesti ed essenziali per la formazione di un laureato in Informatica.

Modalità di verifica dell'apprendimento

L'esame consiste di una prova scritta comprendente una prima parte di domande di teoria e una seconda di esercizi da risolvere, analoghi a quelli assegnati e corretti durante le esercitazioni.

Tipo testo**Risultati di apprendimento attesi*****Testo***

Conoscenza dei principi fondamentali della sintassi e della semantica dei linguaggi di programmazione e delle principali metodologie di parsificazione e traduzione. Conoscenza di strumenti per la progettazione di compilatori di linguaggi di programmazione.

Testi in inglese

Tipo testo

Testo

Lingua insegnamento

Italian

Contenuti

Central concepts of formal languages: alphabet, string, language.
Finite state automata: informal description, deterministic, non-deterministic and with epsilon transitions finite automata. Equivalence of different types of finite state automata and their applications in text searches.
Regular expressions and languages. Applications of regular expressions and their algebraic properties. Properties of regular languages: closure properties and decision problems.
Finite state automata recognizers of regular languages, minimization of automata.
Grammars and context-free languages: parse trees, ambiguities in grammars and languages.
Deterministic and non-deterministic pushdown automata. Pushdown automata recognizers of context-free languages. Properties of context-free languages.
Structure of a compiler.
Top-down parsing: predictive parsing and LL(1) grammars, recursive descent parsing.
Bottom-up parsing: shift-reduce parsing, simple LR parsing.
Syntax-directed translation: inherited and synthesized attributes, evaluation order, syntax-directed translation schemes. Implementation of L-attributed syntax-directed definitions: translation during the recursive descent parsing.
Intermediate code generation: three-address code. Translation of arithmetic expressions, Boolean expressions and translation of statements.

Testi di riferimento

J.E. Hopcroft, R. Motwani and J.D. Ullman, "Introduction to Automata Theory, Languages and Computations", 3rd Edition, Pearson Education, Addison Wesley, 2007.
A.V. Aho, M.S. Lam, R.Sethi, J.D. Ullman. "Compilers: Principles, Techniques and Tools". 2nd Edition, Pearson Education, Addison Wesley, 2007.
Terence Parr, The Definitive ANTLR Reference: Building Domain-Specific Languages, The Pragmatic Bookshelf, 2007, ISBN 978-0978739256.

Obiettivi formativi

The course "Formal Languages and Translators" aims to provide students with an introductory overview of issues related to the definition and translation of programming languages, with particular reference to the design and construction of compilers.
The methodologies and techniques presented in the course are generally useful as general formalism to define the behavior of a system or to implement more simple translators of a compiler itself.

Prerequisiti

The student must be familiar with the basic concepts of set theory and the design of iterative and recursive algorithms. The student must also have acquired the ability to program in high-level languages.

Propedeutico a

The course introduces fundamental notions that are useful in many contexts, and are an essential background for a graduate in Computer Science.

Modalità di verifica dell'apprendimento

The examination consists of a written test, the first part of which comprises questions on theory, and the second part of which comprises exercises similar to those presented in the lectures and teaching materials.

Tipo testo***Testo*****Risultati di
apprendimento attesi**

Knowledge of the basic principles of the syntax of programming languages and the main methods of parsing and translating. Knowledge of tools for the design of compilers for programming languages.

Testi del Syllabus

Docente	ANDRETTA ALESSANDRO	Matricola: 013650
Anno offerta:	2014/2015	
Insegnamento:	MFN0984 - ISTITUZIONI DI LOGICA (Additional Mathematical Logic)	
Corso di studio:	008515 - INFORMATICA (Computer science)	
Anno regolamento:	2014	
CFU:	6	
Settore:	MAT/01	
Tipo attività:	C - Affine/Integrativa	
Partizione studenti:	-	
Anno corso:	1	
Periodo:	Primo Semestre	



Testi in italiano

Tipo testo	Testo
Lingua insegnamento	Italiano
Contenuti	Tecniche di dimostrazione, formalizzazione, principio di induzione, reticoli e ordini, algebre di Boole, insiemi numerabili e più che numerabili
Testi di riferimento	Alberto Facchini, "Algebra e matematica discreta, per studenti di informatica, ingegneria, fisica e matematica con numerosi esempi ed esercizi svolti" Zanichelli/Decibel, 2000.
Obiettivi formativi	Una buona conoscenza dei rudimenti di logica matematica
Prerequisiti	Una ragionevole maturità matematica
Propedeutico a	nessun altro corso
Modalità di verifica dell'apprendimento	Scritto
Risultati di apprendimento attesi	Lo studente sarà in grado di formalizzare nozioni di matematica di base, di utilizzare il principio di induzione e le strutture algebriche su cui si basa la logica matematica



Testi in inglese

Tipo testo

Testo

Lingua insegnamento

Italian

Contenuti

Proof techniques, formalization, induction principle, lattice and orders, Boolean algebras, countable and uncountable sets

Testi di riferimento

Alberto Facchini, "Algebra e matematica discreta, per studenti di informatica, ingegneria, fisica e matematica con numerosi esempi ed esercizi svolti" Zanichelli/Decibel, 2000.

Obiettivi formativi

A solid knowledge of the rudiments of matematica logic

Prerequisiti

A reasonable mathematical maturity

Propedeutico a

no other courses

Modalità di verifica dell'apprendimento

Written test

Risultati di apprendimento attesi

The student will be able to formalize basic mathematical statements, use the principle of induction, and the algebraic structures upon which mathematical logic is built

Testi del Syllabus

Docente	GUNETTI DANIELE	Matricola: 013602
Anno offerta:	2014/2015	
Insegnamento:	MFN1474 - ISTITUZIONI DI SISTEMI OPERATIVI (Additional Operating Systems)	
Corso di studio:	008515 - INFORMATICA (Computer science)	
Anno regolamento:	2014	
CFU:	6	
Settore:	INF/01	
Tipo attività:	D - A scelta dello studente	
Partizione studenti:	-	
Periodo:	Primo Semestre	



Testi in italiano

Tipo testo	Testo
Lingua insegnamento	Italiano
Contenuti	<ul style="list-style-type: none">* Introduzione al Corso di Sistemi Operativi* PARTE I: GENERALITA'<ul style="list-style-type: none">o Introduzione (cap. 1)o Strutture dei Sistemi Operativi (cap. 2)* PARTE II: GESTIONE DEI PROCESSI<ul style="list-style-type: none">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)<ul style="list-style-type: none">o Memoria Centrale (cap. 8)o Memoria Virtuale (cap. 9)* PARTE IV: GESTIONE DELLA MEMORIA SECONDARIA<ul style="list-style-type: none">o Interfaccia del File System (cap. 10)o Realizzazione del File System (cap. 11)o Memoria Secondaria e Terziaria (Gestione dell'Hard disk)
Testi di riferimento	Silberschatz-Galvin-Gagne. Sistemi Operativi - ottava o settima edizione -
Obiettivi formativi	Fornire una conoscenza di base dell'architettura e del funzionamento dei moderni sistemi operativi.
Prerequisiti	Architettura degli Elaboratori; Programmazione I e II
Propedeutico a	nessun corso
Note	nessuna
Modalità di verifica dell'apprendimento	Scritto obbligatorio, orale facoltativo.

Tipo testo**Testo****Risultati di apprendimento attesi**

Lo studente acquisirà la conoscenza dell'architettura e del funzionamento dei moderni sistemi operativi

Frequenza

non obbligatoria

Testi in inglese

Tipo testo	Testo
Lingua insegnamento	Italian
Contenuti	<ul style="list-style-type: none">* Introduction to Operating systems* PART I: GENERALITIES<ul style="list-style-type: none">o Introduction (chap. 1)o Structure of Operating Systems (chap. 2)* PART II: PROCESS MAMAGEMENT<ul style="list-style-type: none">o Processes (chap. 3)o ThreadS (chap. 4)o CPU Scheduling (chap. 5)o Process sincronization (chap. 6)o Deadlock (chap. 7)* PART III: MEMORY ADMINISTRATION<ul style="list-style-type: none">o Central Memory (chap. 8)o Virtual Memory (chap. 9)* PART IV: SECONDARY MEMORY ADMINISTATION<ul style="list-style-type: none">o File System Interface (chap. 10)o File System Implementation (chap. 11)o Secondary and Tertiary memory (chap. 12)
Testi di riferimento	Silberschatz-Galvin-Gagne. Sistemi Operativi - ottava o settima edizione -
Obiettivi formativi	To provide a basic knowledge of the architecture and functioning of modern operating systems.
Prerequisiti	Computer Architecture I; COMputer programming I and II
Propedeutico a	no course
Note	none
Modalità di verifica dell'apprendimento	Written examination, optional oral examination.

Tipo testo**Testo****Risultati di apprendimento attesi**

The student will understand the architecture and functioning of modern Operating systems

Frequenza

not mandatory

Testi del Syllabus

Docente

GUNETTI DANIELE

Matricola: **013602**

Anno offerta:

2014/2015

Insegnamento:

MFN1475 - ISTITUZIONI DI SISTEMI OPERATIVI (Additional Operating Systems)

Corso di studio:

008515 - INFORMATICA (Computer science)

Anno regolamento:

2014

CFU:

12

Settore:

INF/01

Tipo attività:

D - A scelta dello studente

Partizione studenti:

-

Periodo:

Primo Semestre



Testi in italiano

Tipo testo

Testo

Lingua insegnamento

Italiano

Contenuti

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

Tipo testo

Testo

Testi di riferimento

Teoria (TESTO DI RIFERIMENTO):

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

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)

Obiettivi formativi

Fornire una conoscenza di base dell'architettura e del funzionamento dei moderni sistemi operativi. Introdurre all'uso del linguaggio C.

Prerequisiti

Architettura degli Elaboratori; Programmazione I e II

Propedeutico a

nessun corso

Note

nessuna

Modalità di verifica dell'apprendimento

Scritto obbligatorio, sviluppo e discussione di una esercitazione finale, orale facoltativo.

Risultati di apprendimento attesi

Lo studente acquisirà la conoscenza dell'architettura e del funzionamento dei moderni sistemi operativi. Sarà in grado di sviluppare programmi scritti nel linguaggio C, e potrà sviluppare programmi di sistema e concorrenti in ambiente Unix.

Frequenza

Non obbligatoria

Testi in inglese

Tipo testo

Testo

Lingua insegnamento

Italian

Contenuti

Please note: for the theory of the course, the content is based on the referring teaching book.

THEORY:

- * Introduction to Operating systems

- * PART I: GENERALITIES

- o Introduction (chap. 1)

- o Structure of Operating Systems (chap. 2)

- * PART II: PROCESS MAMAGEMENT

- o Processes (chap. 3)

- o ThreadS (chap. 4)

- o CPU Scheduling (chap. 5)

- o Process sincronization (chap. 6)

- o Deadlock (chap. 7)

- * PART III: MEMORY ADMINISTRATION

- o Central Memory (chap. 8)

- o Virtual Memory (chap. 9)

- * PART IV: SECONDARY MEMORY ADMINISTRATION

- o File System Interface (chap. 10)

- o File System Implementation (chap. 11)

- o Secondary and Tertiary memory (chap. 12)

=====

LABORATORY:

- C language

- Introduction to Unix (commands, shell, file system)

- Make and makefile

- System call: processes creation and sincronization

- System call: interprocess communication and signal handling

Practical exercise to develop C programs, to use the Unix system and to write concurrent programs in Unix environment.

Tipo testo

Testo

Testi di riferimento

Theory (REFERRING TEXT):

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

LABORATORY:

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)

Obiettivi formativi

To provide a basic knowledge of the architecture and functioning of modern operating systems. To introduce to the use of C language.

Prerequisiti

Computer Architecture I;
Computer programming I and II

Propedeutico a

no course

Note

none

Modalità di verifica dell'apprendimento

Written examination, development and discussion of a final programming exercise, optional oral examination.

Risultati di apprendimento attesi

The student will understand the architecture and functioning of modern Operating systems. He/She will be able to develop programs written in C language, and will be able to develop cocncurrent programs for the Unix environment.

Frequenza

Not mandatory

Testi del Syllabus

Docente	RADICIONI DANIELE PAOLO	Matricola: 033604
Anno offerta:	2014/2015	
Insegnamento:	MFN0963 - LINGUA INGLESE II (English II)	
Corso di studio:	008515 - INFORMATICA (Computer science)	
Anno regolamento:	2014	
CFU:	3	
Settore:	NN	
Tipo attività:	F - Altro	
Partizione studenti:	-	
Periodo:	Annualità Singola	



Testi in italiano

Tipo testo	Testo
Lingua insegnamento	Inglese
Contenuti	<ul style="list-style-type: none">- Struttura di una lettera e di testi tecnici.- Lessico.- Costruzioni tipiche della scrittura scientifica.- Pratica.
Testi di riferimento	<ul style="list-style-type: none">- New English File Pre-Intermediate, Oxford University Press- Collocations Extra, Cambridge University Press- Sono disponibili dispense redatte dall'esperto linguistico.
Obiettivi formativi	Il corso intende fornire strumenti per la comprensione di articoli scientifici.
Prerequisiti	Conoscenza di base della lingua inglese.
Propedeutico a	La maggior parte della documentazione dei corsi della laurea magistrale è in Inglese ed è perciò richiesta agli studenti conoscenza almeno dell'inglese tecnico.
Modalità di verifica dell'apprendimento	Agli studenti viene proposto un testo tecnico con attinenza all'ambito ICT; essi dovranno dimostrare di comprendere il testo rispondendo correttamente a un questionario. La prova dura un'ora e mezza, ed è permesso l'uso del dizionario (monolingua o bilingue) che ogni studente dovrà portare con se'.
Risultati di apprendimento attesi	Al termine del corso lo studente sarà in grado di leggere un articolo e di rispondere ad alcune brevi domande inerenti il testo letto.
Frequenza	Fortemente consigliata.



Testi in inglese

Tipo testo	Testo
Lingua insegnamento	English
Contenuti	<ul style="list-style-type: none">- Structure of a letter and technical texts.- Lexicon.- Typical constructs of scientific writing.- Practice.
Testi di riferimento	<ul style="list-style-type: none">- New English File Pre-Intermediate, Oxford University Press- Collocations Extra, Cambridge University Press- Handouts prepared by the course Lecturers, available on the course site.
Obiettivi formativi	The course is for developing students' abilities in reading and comprehending scientific papers.
Prerequisiti	Basic knowledge of English.
Propedeutico a	Most documentation of the Computer Science master degree courses is in English; students are therefore requested to be able to read scientific papers and articles.
Modalità di verifica dell'apprendimento	Students must read a text with technical and scientific relevance to ICT; they must demonstrate that they correctly understand the text by answering a questionnaire. The test lasts one hour and a half, and students are allowed to use their own dictionary (monolingual or bilingual proficiency).
Risultati di apprendimento attesi	Students are expected to be able to read scientific articles and answer some simple questions about it.
Frequenza	Attending the course is strongly recommended.

Testi del Syllabus

Docente	RADICIONI DANIELE PAOLO	Matricola: 033604
Anno offerta:	2014/2015	
Insegnamento:	MFN0963 - LINGUA INGLESE II (English II)	
Corso di studio:	008515 - INFORMATICA (Computer science)	
Anno regolamento:	2014	
CFU:	3	
Settore:	NN	
Tipo attività:	F - Altro	
Partizione studenti:	-	
Periodo:	Annualità Singola	



Testi in italiano

Tipo testo	Testo
Lingua insegnamento	Inglese
Contenuti	<ul style="list-style-type: none">- Struttura di una lettera e di testi tecnici.- Lessico.- Costruzioni tipiche della scrittura scientifica.- Pratica.
Testi di riferimento	<ul style="list-style-type: none">- New English File Pre-Intermediate, Oxford University Press- Collocations Extra, Cambridge University Press- Sono disponibili dispense redatte dall'esperto linguistico.
Obiettivi formativi	Il corso intende fornire strumenti per la comprensione di articoli scientifici.
Prerequisiti	Conoscenza di base della lingua inglese.
Propedeutico a	La maggior parte della documentazione dei corsi della laurea magistrale è in Inglese ed è perciò richiesta agli studenti conoscenza almeno dell'inglese tecnico.
Modalità di verifica dell'apprendimento	Agli studenti viene proposto un testo tecnico con attinenza all'ambito ICT; essi dovranno dimostrare di comprendere il testo rispondendo correttamente a un questionario. La prova dura un'ora e mezza, ed è permesso l'uso del dizionario (monolingua o bilingue) che ogni studente dovrà portare con se'.
Risultati di apprendimento attesi	Al termine del corso lo studente sarà in grado di leggere un articolo e di rispondere ad alcune brevi domande inerenti il testo letto.
Frequenza	Fortemente consigliata.



Testi in inglese

Tipo testo	Testo
Lingua insegnamento	English
Contenuti	<ul style="list-style-type: none">- Structure of a letter and technical texts.- Lexicon.- Typical constructs of scientific writing.- Practice.
Testi di riferimento	<ul style="list-style-type: none">- New English File Pre-Intermediate, Oxford University Press- Collocations Extra, Cambridge University Press- Handouts prepared by the course Lecturers, available on the course site.
Obiettivi formativi	The course is for developing students' abilities in reading and comprehending scientific papers.
Prerequisiti	Basic knowledge of English.
Propedeutico a	Most documentation of the Computer Science master degree courses is in English; students are therefore requested to be able to read scientific papers and articles.
Modalità di verifica dell'apprendimento	Students must read a text with technical and scientific relevance to ICT; they must demonstrate that they correctly understand the text by answering a questionnaire. The test lasts one hour and a half, and students are allowed to use their own dictionary (monolingual or bilingual proficiency).
Risultati di apprendimento attesi	Students are expected to be able to read scientific articles and answer some simple questions about it.
Frequenza	Attending the course is strongly recommended.

Testi del Syllabus

Docente	RADICIONI DANIELE PAOLO	Matricola: 033604
Anno offerta:	2014/2015	
Insegnamento:	MFN0963 - LINGUA INGLESE II (English II)	
Corso di studio:	008515 - INFORMATICA (Computer science)	
Anno regolamento:	2014	
CFU:	3	
Settore:	NN	
Tipo attività:	F - Altro	
Partizione studenti:	-	
Periodo:	Annualità Singola	

Testi in italiano

Tipo testo	Testo
Lingua insegnamento	Inglese
Contenuti	<ul style="list-style-type: none">• Struttura di una lettera e di testi tecnici.• Lessico.• Costruzioni tipiche della scrittura scientifica.• Pratica.
Testi di riferimento	<ul style="list-style-type: none">• New English File Pre-Intermediate, Oxford University Press• Collocations Extra, Cambridge University Press• Sono disponibili dispense redatte dall'esperto linguistico.
Obiettivi formativi	Il corso intende fornire strumenti per la comprensione di articoli scientifici.
Prerequisiti	Conoscenza di base della lingua inglese.
Propedeutico a	La maggior parte della documentazione dei corsi della laurea magistrale è in Inglese ed è perciò richiesta agli studenti conoscenza almeno dell'inglese tecnico.
Modalità di verifica dell'apprendimento	Agli studenti viene proposto un testo tecnico con attinenza all'ambito ICT; essi dovranno dimostrare di comprendere il testo rispondendo correttamente a un questionario. La prova dura un'ora e mezza, ed è permesso l'uso del dizionario (monolingua o bilingue) che ogni studente dovrà portare con se'.
Risultati di apprendimento attesi	Al termine del corso lo studente sarà in grado di leggere un articolo e di rispondere ad alcune brevi domande inerenti il testo letto.
Frequenza	Fortemente consigliata.



Testi in inglese

Tipo testo	Testo
Lingua insegnamento	English
Contenuti	<ul style="list-style-type: none">• Structure of a letter and technical texts.• Lexicon.• Typical constructs of scientific writing.• Practice.
Testi di riferimento	<ul style="list-style-type: none">- New English File Pre-Intermediate, Oxford University Press- Collocations Extra, Cambridge University Press- Handouts prepared by the course Lecturers, available on the course site.
Obiettivi formativi	The course is for developing students' abilities in reading and comprehending scientific papers.
Prerequisiti	Basic knowledge of English.
Propedeutico a	Most documentation of the Computer Science master degree courses is in English; students are therefore requested to be able to read scientific papers and articles.
Modalità di verifica dell'apprendimento	Students must read a text with technical and scientific relevance to ICT; they must demonstrate that they correctly understand the text by answering a questionnaire. The test lasts one hour and a half, and students are allowed to use their own dictionary (monolingual or bilingual proficiency).
Risultati di apprendimento attesi	Students are expected to be able to read scientific articles and answer some simple questions about it.
Frequenza	Attending the course is strongly recommended.

Testi del Syllabus

Docente

SEMPLICE MATTEO

Matricola: **613347**

Anno offerta:

2014/2015

Insegnamento:

MFN0962 - METODI NUMERICI (Numerical Methods)

Corso di studio:

008515 - INFORMATICA (Computer science)

Anno regolamento:

2014

CFU:

6

Settore:

MAT/08

Tipo attività:

C - Affine/Integrativa

Partizione studenti:

-

Anno corso:

1

Periodo:

Primo Semestre

Tipo testo	Testo
Lingua insegnamento	italiano
Contenuti	<p>1. Nozioni introduttive. Rappresentazione dei numeri e sistemi numerici. Aritmetica in virgola mobile, errori di rappresentazione e loro propagazione. Numeri di condizionamento.</p> <p>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.</p> <p>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.</p> <p>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.</p> <p>5. Equazioni non lineari. Metodi di bisezione, di punto fisso, di Newton e principali varianti. Metodi di globalizzazione (line search). Minimizzazione.</p> <p>6. Equazioni differenziali ordinarie. Definizioni ed esempi. Metodi numerici espliciti ed impliciti (Runge-Kutta).</p> <p>LABORATORIO Analisi di algoritmi numerici e soluzione numerica di problemi con l'uso del software scientifico MATLAB.</p>
Testi di riferimento	<p>Libri consigliati</p> <p>Quarteroni, Saleri, Gervasio - "Calcolo Scientifico: esercizi e problemi risolti con MatLab e Octave" - Springer (Also available in english as "Scientific Computing with MatLab and Octave")</p> <p>V. Comincioli - Metodi numerici e statistici per le scienze applicate - http://www.multimediacampus.it/offerta/risorse.htm</p>
Obiettivi formativi	<p>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, insegnando tecniche generali utili nel trattamento dell'informazione numerica tramite calcolatore.</p> <p>Il corso è completato dall'attività di laboratorio, in cui viene utilizzato il software scientifico MATLAB, per la soluzione di problemi numerici e programmare gli algoritmi introdotti a lezione.</p>
Prerequisiti	<ul style="list-style-type: none">- analisi di funzioni reali di variabile reale (massimi, minimi, formula di Taylor)- algebra lineare (moltiplicazione riga per colonna, autovalori di una matrice quadrata, sistemi lineari)

Tipo testo

Testo

Modalità di verifica dell'apprendimento

L'esame scritto si svolge in laboratorio, con a disposizione il software MatLab, L'esame orale è facoltativo.

Risultati di apprendimento attesi

Al termine del corso lo studente avrà

- appreso i più semplici algoritmi dei principali capitoli dell'analisi numerica e avrà le basi teoriche per procedere autonomamente nello studio di quelli più avanzati
- imparato a programmare nell'ambiente MatLab
- imparato a riconoscere ed evitare i più comuni errori che possono insorgere nel trattare variabili di tipo numerico con il calcolatore



Testi in inglese

Tipo testo

Testo

Lingua insegnamento

italian

Contenuti

1. Introduction.

Number representation and machine number systems. Floating point arithmetics, errors and their propagation. Condition numbers.

2. Numerical linear algebra.

Review of linear algebra, vector and matrix norms, condition number of a matrix. Gauss elimination, matrix factorization. Iterative methods for linear systems (Jacobi, Gauss-Seidel). Eigenvalues and eigenvectors: localization theorems and power method.

3. Interpolation and approximation.

Polynomial interpolation. Lagrange and Newton formulas. Interpolation error, Chebichev points, convergence. Piece-wise polynomial interpolants (splines). Trigonometric interpolation.

Approximation. Least squares method. Linear regression. Linearization of data.

4. Numerical quadrature.

Interpolatory quadrature rules with equispaced nodes (trapezoidal, Cavalieri-Simpson, midpoint) and optimal ones (gaussian). Composite quadrature rule and automatic quadrature.

5. Non linear equazioni.

Methods of bisection, fixed point iterations, Newton and main variants. Globalization methods (line search). Search of local extrema.

6. Ordinary differential equations.

Definition and examples. Explicit and implicit numerical methods (Runge-Kutta).

IT LABORATORY

Analysis of numerical algorithms and numerical solution of problems with the MATLAB software.

Testi di riferimento

Suggested books

Quarteroni, Saleri, Gervasio - "Calcolo Scientifico: esercizi e problemi risolti con MatLab e Octave" - Springer (Also available in english as "Scientific Computing with MatLab and Octave")

V. Comincioli - Metodi numerici e statistici per le scienze applicate - <http://www.multimediacampus.it/offerta/risorse.htm>

Obiettivi formativi

This course concerns techniques for the trustworthy and efficient numerical solution of mathematical problems and for the interpretation of the results. Particular attention is given to the analysis of errors and to the computational aspects, teaching general techniques that are useful in the treatment of numerical information with a computer.

Part of the course will be held in a computing lab, where the scientific computing software MatLab is employed to solve numerical problems and test the algorithms studied in the lectures.

Prerequisiti

- real analysis (local extrema, Taylor formula)

- linear algebra (matrix multiplication rule, eigenvalues of a square matrix, linear systems)

Tipo testo

Testo

Modalità di verifica dell'apprendimento

The written examination takes place in the computing lab, with the possibility of using the MatLab software. The viva is not compulsory.

Risultati di apprendimento attesi

At the end of the course the student will

- know the simplest algorithms belonging to the most relevant chapters of numerical analysis and possess the theoretical background needed to progress on its own towards the more advanced ones
- be able to program in MatLab
- recognize and avoid the most common pitfalls in the treatment of numerical quantities with a computer

Testi del Syllabus

Docente	BALOSSINO NELLO	Matricola: 010522
Anno offerta:	2014/2015	
Insegnamento:	MFN0973 - MODELLAZIONE GRAFICA (Graphic modeling)	
Corso di studio:	008515 - INFORMATICA (Computer science)	
Anno regolamento:	2014	
CFU:	9	
Settore:	INF/01	
Tipo attività:	B - Caratterizzante	
Partizione studenti:	-	
Anno corso:	1	
Periodo:	Secondo Semestre	



Testi in italiano

Tipo testo

Testo

Lingua insegnamento

Italiano

Contenuti

Una panoramica sulla computer grafica.
Matematica per la computer grafica.
Formati di file grafici.
Equazione del piano, distanza punto-piano, angolo tra piani, faccia anteriore e posteriore di un poligono. Primitive grafiche di output. Sistemi di riferimento di coordinate. Algoritmi per il tracciamento di linee: equazioni lineari, algoritmo DDA, algoritmo di Bresenham per le linee. Esempi. Algoritmi per la generazione di circonferenze e ellissi. Algoritmi Midpoint per ellissi e circonferenze. Esempi.
Trasformazioni geometriche bidimensionali: traslazione, rotazione, scalamento. Rappresentazione matriciale e coordinate omogenee. Trasformazioni inverse.
Composizione di trasformazioni geometriche bidimensionali: traslazione, scalamento, rotazione attorno a un punto, scalamento rispetto a un punto. Scalamento bidimensionale generico in due direzioni.
Altre trasformazioni bidimensionali: riflessione, taglio. Trasformazioni tra sistemi di coordinate bidimensionali.
Visualizzazione bidimensionale; la pipeline per la visualizzazione bidimensionale, la finestra di clipping. Trasformazioni di normalizzazione e viewport. Il mapping della finestra di clipping nella viewport normalizzata e nel quadrato. Algoritmi di clipping: clipping bidimensionale di punti e linee. Clipping di Cohen-Sutherland. Clipping di poligoni con Sutherland-Hodgman.
Trasformazioni geometriche tridimensionali: traslazione in 3D, rotazioni in 3D. Rotazione di assi coordinati in 3D. scalamento in 3D.
Rotazioni in 3D generiche. Metodo dei quaternioni per rotazioni in 3D.
Composizione di trasformazioni geometriche tridimensionali
Altre trasformazioni tridimensionali: riflessioni in 3D, taglio in 3D.
Trasformazioni tra sistemi di coordinate tridimensionali.
Trasformazioni affini.
Visualizzazione di spazi tridimensionali: la pipeline per la visualizzazione tridimensionale. Trasformazioni di parametri per la visualizzazione tridimensionale: da coordinate mondiali a quelle di visualizzazione, trasformazione di proiezione, proiezioni ortogonali, proiezioni parallele oblique. Proiezioni prospettiche: trasformazione di coordinate, punto di fuga, volume di visualizzazione. Frustum simmetrico per la proiezione prospettica. Frustum per la proiezione prospettica obliqua. Coordinate normalizzate per la trasformazione di proiezione prospettica.
La trasformazione di viewport e coordinate di visualizzazione tridimensionali. Algoritmi di clipping in 3D.
Rappresentazione di oggetti tridimensionali: superfici quadratiche, rappresentazione di spline. Metodi di interpolazione di spline cubiche: interpolazione di Hermite; curve di Bezier: equazioni e proprietà. Curve cubiche di Bezier. Curve B-spline periodiche. Superfici B-spline. Spline razionali.
Quadtree, octree, alberi BSP.
Metodi di geometria frattale: procedure di generazione di oggetti mediante frattali, classificazione dei frattali, dimensione frattale, costruzione geometrica di frattali deterministici autosimili.
Pattern di retinatura e tecniche. Tessitura e metodi per il dettaglio delle superfici.
Modelli 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 oggetti articolati. Motion capture. Movimenti periodici.
Radiometria e fotometria.
Lezioni di modellazione con Blender.

Tipo testo

Testo

Testi di riferimento

Donald D. Hearn, M. Pauline Baker, Warren Carithers, "Computer Graphics with Open GL (4th Edition)", Pearson Education, 2011

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, 2014

Obiettivi formativi

È fornita una conoscenza dettagliata delle tecniche atte a creare oggetti geometrici bi- e tri-dimensionali e a utilizzare opportuni modificatori. La costruzione di una scena di sintesi ha infatti come obiettivo sia la definizione della geometria degli oggetti che la compompono, sia la possibilità di poterne variare l'aspetto, a seconda del contesto di utilizzo, nonché di osservarli da un qualsiasi punto di vista. È pertanto necessario acquisire padronanza della terminologia e delle tecniche analitiche che regolano i passi di creazione e modellazione mediante sistemi automatici. La conversione dei contenuti teorici in software applicativo, mediante un programma di modellazione (Blender), fornisce allo studente una visione completa del processo di sintesi.

Prerequisiti

Conoscenza delle nozioni fornite dai corsi di matematica, con particolare riferimento al calcolo matriciale e alla geometria analitica.

Propedeutico a

Preparazione della prova finale in una visione interdisciplinare.

Note

Sono forniti dal docente appunti sugli argomenti di parte delle lezioni.

Modalità di verifica dell'apprendimento

Durante il corso vengono proposti agli studenti problemi sui temi trattati e discusse con loro possibili soluzioni. L'esame consiste nella realizzazione di un progetto in Blender e un colloquio orale su tutti gli argomenti trattati nel corso.

Risultati di apprendimento attesi

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

Frequenza

La frequenza è consigliata ma non è obbligatoria.



Testi in inglese

Tipo testo

Testo

Lingua insegnamento

Italian

Contenuti

A survey of Computer Graphics.
Mathematics for Computer Graphics.
Graphics File Formats.
Plane equations, point-plane distance, angle between planes, front and back polygon faces. Graphics output primitives: Coordinates references frames, Line Drawing Algorithms: linear equations, DDA algorithm, Bresenham's Line Algorithm, Examples. Circle and Ellipse generating algorithms, Midpoint Circle and Ellipse Algorithms, Examples.
Two Dimensional Geometric Transformations: Translation, Rotation, Scaling. Matrix Representation and Homogeneous Coordinates. Inverse Transformations.
Two Dimensional Composite Transformations: translations, scaling, pivot point Rotation, fixed point Scaling. General Two Dimensional Scaling Directions.
Other Two Dimensional Transformations: Reflection, Shear. Transformation between Two Dimensional Coordinate Systems.
Two Dimensional Viewing: Two Dimensional Viewing Pipeline, the Clipping Window. Normalization and Viewport Transformations: Mapping the Clipping Window into Normalized Viewport and Square. Clipping Algorithms: Two Dimensional Point and Line Clipping. Cohen-Sutherland Clipping. Sutherland-Hodgman Polygon Clipping.
Geometric Transformations in Three Dimensional Space. Three Dimensional Translation. Three Dimensional Rotations. Three Dimensional Coordinates Axis Rotations, General Three Dimensional Rotations. Quaternion Methods for Three Dimensional Rotations. Three Dimensional Scaling. Composite Three Dimensional Transformations.
Other Three Dimensional Transformations, Three Dimensional Reflections, Three Dimensional Shear. Transformations between Three Dimensional Coordinates Systems. Affine Transformations.
Three Dimensional Viewing: the Three Dimensional Viewing Pipeline. Three Dimensional Viewing Coordinates Parameters Transformation from World to Viewing Coordinates, Projection Transformations, Orthogonal Projections. Oblique Parallel Projections. Perspective Projections: Coordinates Transformation, Vanishing Points, View volume. Symmetric Perspective-Projection Frustum. Oblique Perspective-Projection Frustum. Normalized Perspective-projection Transformation Coordinates.
The Viewport Transformation and Three Dimensional Screen Coordinates. Three Dimensional Clipping Algorithms.
Three dimensional object representations: quadratic surfaces, Spline representation. Cubic spline interpolation methods: Hermite interpolation. Bezier spline curves: Bezier curve equations, properties of Bezier curves, cubic Bezier curves. Cubic periodic B-spline curves. B-spline surfaces. Rational splines.
Quadtree, Octree, BSP-trees.
Fractal Geometry methods: fractal generation procedures, classification of fractals, fractal dimension, geometric construction of deterministic self-similar fractals.
Halftone Patterns and Dithering Techniques. Texturing and Surface Detail Methods.
Color Models and Color Applications. Bump mapping.
Computer animation. Design of animation sequences. Traditional animation sequences. Key-Frame Systems. Character Animation: Articulated Figure Animation, Motion Capture. Periodic Motions.
Radiometry and photometry.
Lessons on modeling with Blender.

Tipo testo

Testo

Testi di riferimento

Donald D. Hearn, M. Pauline Baker, Warren Carithers, "Computer Graphics with Open GL (4th Edition)", Pearson Education, 2011

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, 2014

Obiettivi formativi

The student is supplied with a detailed knowledge of techniques aimed at building 2D and 3D geometric objects using suitable modifiers. Indeed, to build a synthetic scene requires defining the geometry of the objects it is composed, leaving the possibility to change their appearance according to the context, and the ability to observe them from every point.

It is thus necessary to acquire the terminology and the mathematical techniques governing the creation and modeling steps applied in automatic systems.

The implementation of the theoretical concepts into a software application, through a modeling software (Blender), allows the student to have a complete vision of the synthesis process.

Prerequisiti

Previously acquired knowledge from the mathematical courses, in particular operations on matrices and analytic geometry.

Propedeutico a

Preparation for the final exam having an interdisciplinary culture.

Note

Apart from the suggested textbooks, material from the teacher covering part of the lessons is available.

Modalità di verifica dell'apprendimento

During the lessons the teacher proposes problems related to the arguments presented and possible solutions.

The exam consists of the preparation of a project (using the software Blender) and an oral discussion on all the topics presented in the course.

Risultati di apprendimento attesi

Knowledge of the main and advanced techniques to model artificial environments.

Frequenza

Attendance to the lessons is suggested but not mandatory.

Testi del Syllabus

Docente	SACCO GIOVANNI	Matricola: 013423
Anno offerta:	2014/2015	
Insegnamento:	MFN0940 - MODELLI AVANZATI E ARCHITETTURE DI BASI DI DATI (Advanced Data Models and Database Architectures)	
Corso di studio:	008515 - INFORMATICA (Computer science)	
Anno regolamento:	2014	
CFU:	9	
Settore:	INF/01	
Tipo attività:	B - Caratterizzante	
Partizione studenti:	-	
Anno corso:	1	
Periodo:	Primo Semestre	



Testi in italiano

Tipo testo

Testo

Lingua insegnamento

Italiano

Contenuti

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 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

Testi di riferimento

Libri di testo

Modelli Avanzati

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

Architetture

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

Modelli Avanzati

Kent, "Limitations of Record Based Information Models", ACM TODS 4:1, 1979

Architetture

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

Tipo testo

Testo

Comer, The ubiquitous B-tree, ACM CSUR, 11:2, 1979
Enbody, Du, Dynamic hashing schemes, ACM CSUR, 20:2, 1988

Obiettivi formativi

Comprensione delle problematiche e delle soluzioni di implementazione ed architetturali per sistemi database centralizzati e distribuiti, con speciale riferimento ai sistemi database relazionali. Implicazioni sulla performance ed il disegno fisico.

Prerequisiti

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

Note

Il corso di 6 crediti comprende la sola parte di Architetture

Modalità di verifica dell'apprendimento

esame scritto

Risultati di apprendimento attesi

Saper utilizzare i modelli object-oriented e object-relational, avere competenze per la selezione/implementazione di sistemi di information retrieval (motori di ricerca), e di ricerca esplorativa (dynamic taxonomies).

Conoscenze approfondite sull'architettura dei sistemi database e di tecniche di gestione dati in generale, utili sia per il disegno fisico di database, sia per realizzazioni ad-hoc ad alta performance

Frequenza

Non obbligatoria, ma fortemente consigliata



Testi in inglese

Tipo testo

Testo

Lingua insegnamento

Italian

Contenuti

Advanced data models

Object-oriented and object-relational models
Information Retrieval
Dynamic Taxonomies and Faceted Search

Database architectures

Storage devices (includes RAID and Solid State Disks).
Operating systems: disk scheduling, disk allocation, file systems.
Implementation of operations through sequential scans:
selection, projection, join through nested loops, nested scans, merging
scans, fragmentation aka recursive hash partitioning
Semijoins and filtering through binary and Bloom filters. Other
applications of Bloom filters
Buffer management
Indices.
B-trees, B+trees and prefix B-trees. Key compression. Buffering
Hashing. Static methods. Dynamic methods: extendible hashing. Tries.
Secondary indices.
Multiple and inverted lists. Index architectures for information retrieval
and dynamic taxonomies
Query optimization.
Non-uniform access distributions: Zipf and 80-20
Database physical design
Transactions.
Recovery
Concurrency control

Testi di riferimento

Textbooks

Advanced data models

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

Database architectures

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

Papers

Advanced data models

Kent, "Limitations of Record Based Information Models", ACM TODS 4:1,
1979

Database architectures

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

Tipo testo**Testo****Obiettivi formativi**

An in-depth, working knowledge of advanced models for structured (object-oriented and object-relational), semi-structured (dynamic taxonomies) and unstructured (information retrieval) databases.

Understanding the problems and implementation and architectural solutions for centralized and distributed databases, with a special focus on relational database systems, and implications on performance and physical design.

Prerequisiti

Good knowledge of the relational model, Extended ER, SQL2, operating systems and algorithms.

Note

The 6 credit course includes the "Architectures" module only.

Modalità di verifica dell'apprendimento

written examination
(in English by request)

Risultati di apprendimento attesi

An in-depth, working knowledge of advanced models for structured (object-oriented and object-relational), semi-structured (dynamic taxonomies) and unstructured (information retrieval) databases.

Understanding the problems and implementation and architectural solutions for centralized and distributed databases, with a special focus on relational database systems, and implications on performance and physical design.

Frequenza

Not compulsory but highly recommended.

Testi del Syllabus

Docente	MARGARIA INES MARIA	Matricola: 010621
Anno offerta:	2014/2015	
Insegnamento:	MFN0960 - MODELLI CONCORRENTI E ALGORITMI DISTRIBUITI (Concurrent Models and Distributed Algorithms)	
Corso di studio:	008515 - INFORMATICA (Computer science)	
Anno regolamento:	2014	
CFU:	6	
Settore:	INF/01	
Tipo attività:	B - Caratterizzante	
Partizione studenti:	-	
Periodo:	Primo Semestre	



Testi in italiano

Tipo testo	Testo
Lingua insegnamento	Italiano
Contenuti	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.
Testi di riferimento	Ancilotti - Boari. Programmazione concorrente e distribuita. McGRAW-HILL Ben-Ari - Principles of Concorrente 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
Obiettivi formativi	Il corso si propone di fornire le metodologie e gli strumenti per la Programmazione Concorrente e Distribuita, ponendo l'enfasi soprattutto sui vari modelli cui i linguaggi fanno riferimento e sugli strumenti tipici di ogni modello. Lo studio di questi modelli riguarda principalmente i costrutti per esprimere la concorrenza dei processi e le loro possibili interazioni. L'analisi viene effettuata adottando 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. Un secondo obiettivo è lo studio e l'analisi di algoritmi distribuiti. Tali algoritmi vengono illustrati e studiati mediante appropriati modelli formali. Questa seconda parte viene affrontata con una serie di seminari.
Prerequisiti	Nessun prerequisito
Propedeutico a	Non sono previste propedeuticità che rendano obbligatorio il superamento del corso.
Modalità di verifica dell'apprendimento	Prova scritta e orale. La partecipazione all'attività seminariale prevede una valutazione con la possibilità di esonero della parte corrispondente.

Testi in inglese

Tipo testo	Testo
Lingua insegnamento	Italian
Contenuti	<p>Concurrent programming: concurrent processes, architecture of a concurrent machine</p> <p>Concurrent programming primitives. Shared memory model: semaphores, monitors, path expressions. Distributed memory model: asynchronous send-receive, synchronous send-receive, rendez-vous, remote procedure call.</p> <p>Coordination in distributed systems: synchronous and asynchronous models, logical time for asynchronous networks.</p> <p>Distributed algorithms: leader election algorithms, mutual exclusion algorithms, deadlock detection algorithms, termination algorithms, consensus problem algorithms.</p>
Testi di riferimento	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
Obiettivi formativi	<p>The course aims to provide methodologies and tools for concurrent and distributed programming, emphasizing especially on the various models the languages refer to.</p> <p>The study of these models is primarily concerned with the constructs to express concurrency and interaction between processes. The analysis is performed by adopting a set of examples that cover a significant area of applications and comparing the solutions within the various models.</p> <p>A second objective is the introduction of distributed algorithms ; the main distributed algorithms are introduced and analyzed by means a suitable theoretic model.</p>
Prerequisiti	No prerequisite is requested.
Propedeutico a	It is not a mandatory pre-requisite to any course.
Modalità di verifica dell'apprendimento	<p>Written text and oral examination.</p> <p>Participation in the seminars includes an assessment which allows exemption from the oral examination of the corresponding part.</p>

Testi del Syllabus

Docente	GROSSO ANDREA CESARE	Matricola: 014296
Anno offerta:	2014/2015	
Insegnamento:	MFN1349 - OTTIMIZZAZIONE COMBINATORIA (Combinatorial Optimization)	
Corso di studio:	008515 - INFORMATICA (Computer science)	
Anno regolamento:	2014	
CFU:	6	
Settore:	MAT/09	
Tipo attività:	C - Affine/Integrativa	
Partizione studenti:	-	
Periodo:	Secondo Semestre	



Testi in italiano

Tipo testo	Testo
Lingua insegnamento	Italiano
Contenuti	Richiami di Ricerca Operativa: dal problema reale agli algoritmi di risoluzione. Esempi di modelli matematici. Richiami di PL. Nozioni di complessità : classi P e NP, problemi NP-completi, approssimazione. Problemi PLI: rilassamenti lineari e chiusure convesse. Matrici totalmente unimodulari. Branch-and-bound: principi generali e applicazioni. Problemi di flusso su reti: modellazione ed algoritmi di soluzione. Programmazione Dinamica. Algoritmi euristici e ricerca locale.
Testi di riferimento	Appunti forniti dal docente. C.H. Papadimitriou, K.R. Steiglitz, "Combinatorial optimization: algorithms and complexity", Prentice Hall
Obiettivi formativi	Familiarizzare lo studente con problemi di ottimizzazione che occorrono frequentemente in applicazioni pratiche, permettendogli di riconoscere la difficoltà del problema (attraverso alcune nozioni di complessità) 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.
Prerequisiti	Ricerca Operativa, con particolare riferimento alla programmazione lineare e alla teoria della dualità
Propedeutico a	--
Modalità di verifica dell'apprendimento	Esame orale e scritto.
Risultati di 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.

Testi in inglese

Tipo testo	Testo
Lingua insegnamento	Italian
Contenuti	Methodology of Operations Research: from the problem to the solution algorithm. Examples of mathematical programming models. Basic of PL. Complexity theory: P and NP classes, NP-complete problems, approximation. PLI: linear relaxations and convex hulls. Unimodularity. Branch-and-bound: basics and applications. Network flow problems: models and solution algorithms. Dynamic programming. Heuristic algorithms and local search.
Testi di riferimento	Lecture notes provided by the instructor. C.H. Papadimitriou, K.R. Steiglitz, "Combinatorial optimization: algorithms and complexity", Prentice Hall
Obiettivi formativi	Acquiring knowledge about fundamental problems in combinatorial optimization that frequently arise in practice (possibly as sub-problems of more complex situations). Being able to deploy algorithmic techniques to solve such problems to optimality or by approximation.
Prerequisiti	Basics in Operations Research, particularly linear programming and duality.
Propedeutico a	--
Modalità di verifica dell'apprendimento	Written and oral exams.
Risultati di apprendimento attesi	The student should be able to recognize combinatorial optimization problems, establish their complexity, choose and implement the appropriate algorithmic techniques for solving such problems.

Testi del Syllabus

Docente	RUFFO GIANCARLO FRANCESCO Matricola: 014114
Anno offerta:	2014/2015
Insegnamento:	MFN0954 - RETI COMPLESSE (Complex Networks)
Corso di studio:	008515 - INFORMATICA (Computer science)
Anno regolamento:	2014
CFU:	6
Settore:	INF/01
Tipo attività:	B - Caratterizzante
Partizione studenti:	-
Periodo:	Primo Semestre

Tipo testo	Testo
Lingua insegnamento	Italiano
Contenuti	<p>Introduzione alle reti complesse: esempi dalla fisica, biologia, sociologia, economia, spettacolo ed informatica.</p> <p>Esempi: caratteristiche di Internet, network degli attori, etc. Ripasso di teoria dei grafi. Definizione di grafo. Grafi semplici e bipartiti. Grafi pesati e non. Grafi diretti e non.</p> <p>Caratterizzazione della topologia di un network a livello globale e di singolo nodo: distribuzione dei parametri per singolo nodo (connettività'). Connettività', clustering, misure di centralità'.</p> <p>Diametro di un network. Sottonetworks; clustering di un network; cliques e moduli. Definizione e calcolo delle principali misure del network. Algoritmi per l'identificazione di comunità.</p> <p>Modello base: random networks alla Erdos-Renyi. Distribuzione dei parametri del network e teoremi limite per reti di grandi dimensioni. Matrici di Wigner e spettro degli autovalori. Transizione di fase e giant cluster. Relazione tra vari parametri del network (assortativity, connettività' vs. betweenness centrality).</p> <p>Generalizzazione a small world networks: high clustering short distances ($\log(N)$). Modello di Watts-Strogatz: rewiring.</p> <p>Scale free networks: esempi. Modello di crescita di Barabasi-Alberts: preferential attachment. Scale log-log. Istogrammi e possibili errori (heavy tails).</p> <p>Casi di Studio: Social Media, Sistemi P2P; Reti di similarità' ed applicazioni nei sistemi di recommendation; fenomeni ed algoritmi epidemici; Applicazioni di reti sociali e comunità' virtuali; tecniche di visualizzazione.</p>
Testi di riferimento	<p>M. E. J. Newman, Networks: An Introduction, Oxford University Press, Oxford (2010)</p> <p>D. Easley and J. Kleinberg, Networks, Crowds, and Markets: Reasoning About a Highly Connected World, Cambridge University Press (2010)</p>
Obiettivi formativi	<p>Questo corso si concentra su sistemi complessi che possono essere descritti nei termini di una rete, dove gli elementi sono dei nodi (o vertici di un grafo) e le interazioni sono i collegamenti (o archi). La Scienza della Complessità', in questo ambito, basandosi su solidi principi teorici, fornisce un vasto insieme di strumenti e di metodologie che possono essere sfruttati in contesti applicativi diversi: ad esempio, per individuare criticità' nelle reti tecnologiche, per ottimizzare i servizi di recommendation e viral marketing nei sistemi di social media, per tracciare il comportamento degli attori maggiormente influenti in una rete sociale, o per studiare la propagazione di un'epidemia a livello globale.</p> <p>Le tecniche che saranno introdotte sono utilizzate in ambito multidisciplinare da sociologi, fisici, biologi, epidemiologi e - ovviamente - informatici. Ci concentreremo soprattutto sugli aspetti algoritmici e sui modelli, senza tralasciare l'applicazione degli strumenti appresi attraverso tool di analisi appropriati. A questo scopo, infatti, saranno proposte sessioni di esercitazione pratica su dati fittizi e reali, che possono essere rappresentati graficamente per mezzo di tool avanzati progettati per la visualizzazione di dati complessi.</p>
Prerequisiti	Basi di teoria dei grafi; statistica; algebra lineare; reti.
Propedeutico a	laurea magistrale

Tipo testo

Testo

Modalità di verifica dell'apprendimento

Verranno svolte esercitazioni periodiche, spesso sfruttando dati estratti da casi reali per individuare pattern ed eventuali differenze caratterizzanti rispetto ad altre reti. Esame orale e/o progetto finale.

Risultati di apprendimento attesi

Analisi di reti complesse; Uso di tool di analisi esistenti; Applicazioni in ambito di social networking e web analysis e mining. Comprensione di alcuni fattori causali dei fenomeni che si individuano nelle reti complesse. Uso di tool di visualizzazione di strutture dati complesse.

Testi in inglese

Tipo testo

Testo

Lingua insegnamento

Italiano

Contenuti

Introduction to complex networks: examples in physics, biology, sociology, economy, entertainment and computer science.
Case studies: Internet, actor-movies graph, and so on.
Graph theory survey. Definition of a graph. Simple graphs and bipartite networks. Weighted graphs. Undirected and directed graph.
Networks topology and (local and global) structure: degree distribution, connectivity. Components. Centrality measures.
Diameter of a network. Cliques and subgraphs. Definition and computation of the main measures of a network. Community detection.
Models of a network: Erdos-Renyi's random graphs. Parameters distribution and limit theorem for large scale networks. Wigner's matrices and spectral analysis. Phase transition and giant component.
Relationships between different parameters in a network (assortativity, connectivity vs. betweenness centrality).
Small world models: high clustering and short distances ($\log(n)$). Watts-Strogatz model: rewiring.
Scale free networks: case studies. Growing model of Barabasi-Alberts: preferential attachment. Log-log scale. Histograms and statistical error. Heavy tail distributions.
Case studies: Social Media, P2P systems; Similarity networks and recommendation systems; epidemics; Social network and virtual communities; Visualization techniques.

Testi di riferimento

M. E. J. Newman, Networks: An Introduction, Oxford University Press, Oxford (2010)
D. Easley and J. Kleinberg, Networks, Crowds, and Markets: Reasoning About a Highly Connected World, Cambridge University Press (2010)

Testi del Syllabus

Docente

GARETTO MICHELE

Matricola: **014400**

Anno offerta:

2014/2015

Insegnamento:

MFN0968 - RETI II (Computer Networks II)

Corso di studio:

008515 - INFORMATICA (Computer science)

Anno regolamento:

2013

CFU:

9

Settore:

INF/01

Tipo attività:

B - Caratterizzante

Partizione studenti:

-

Anno corso:

2

Periodo:

Primo Semestre

Tipo testo	Testo
Lingua insegnamento	Italiano
Contenuti	<p>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 architettoniche. Esamineremo i pro e i contro della architettura attuale, e rifletteremo su come rendere Internet migliore in futuro.</p> <p>Argomenti del corso</p> <ul style="list-style-type: none">-Principi fondamentali: architettura end-to-end, nomi e indirizzi, segnalazione, segmentazione, randomizzazione, indirectione, 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.
Testi di riferimento	<p>Reti di calcolatori e internet. Un pproccio top-down, 6/Ed. James F. Kurose, Keith W. Ross, Pearson Education Italia, 2013</p> <p>Communication Networks An Optimization, Control and Stochastic Networks Perspective, R. Srikant, Lei Ying, Cambridge University Press, 2014</p>
Obiettivi formativi	<p>Gli obiettivi del corso sono:</p> <ul style="list-style-type: none">- 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.
Prerequisiti	Reti di elaboratori
Modalità di verifica dell'apprendimento	Esame scritto e presentazione orale
Frequenza	Facoltativa



Testi in inglese

Tipo testo	Testo
Lingua insegnamento	Italian
Contenuti	<p>In the past two decades, the Internet has grown from a research tool to a fundamental part of society; something we all take for granted and use everyday. In this class we'll explore why the infrastructure was designed this way, and the underlying principles and design decisions. We'll examine the pros and cons of the current design, and give some thought to how we can make the Internet better in future.</p> <p>Topics</p> <ul style="list-style-type: none">-Fundamental principles: end-to-end design, naming and addressing, signalling, segmentation, randomization, indirection, multiplexing, virtualization, scalability.-Network resource allocation: traffic engineering, congestion control as a resource allocation problem, TCP.-Router design: input-output-queueing, packet classification, scheduling.-Content Centric Networks: content distribution networks, names vs. addresses, network caches.-BitTorrent: protocol specifications, performance modeling.-Cellular networks: architecture and principles, evolution of the standards.-Sensor networks: basic principles and main issues.-Wireless networking: complements on 802.11, Bluetooth, ad-hoc networks, opportunistic routing and scheduling.
Testi di riferimento	<p>Computer Networking: A Top-Down Approach, 6/E James F. Kurose, Keith W. Ross, Addison-Wesley, 2013</p> <p>Communication Networks An Optimization, Control and Stochastic Networks Perspective, R. Srikant, Lei Ying, Cambridge University Press, 2014</p>
Obiettivi formativi	<p>The goals for this class are:</p> <ul style="list-style-type: none">- To become familiar with the state of the art in networking research: network architecture, protocols and systems.- To gain some practice in reading research papers and critically understand them.- To learn how to effectively present and discuss in public a networking topic.
Prerequisiti	Computer Networks
Modalità di verifica dell'apprendimento	Written test and oral presentation
Frequenza	Optional

Testi del Syllabus

Docente	GARETTO MICHELE	Matricola: 014400
Anno offerta:	2014/2015	
Insegnamento:	INF0009 - RETI II (Computer Networks II)	
Corso di studio:	008515 - INFORMATICA (Computer science)	
Anno regolamento:	2014	
CFU:	6	
Settore:	INF/01	
Tipo attività:	B - Caratterizzante	
Partizione studenti:	-	
Anno corso:	1	
Periodo:	Primo Semestre	

Tipo testo	Testo
Lingua insegnamento	Italiano
Contenuti	<p>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 architettoniche. Esamineremo i pro e i contro della architettura attuale, e rifletteremo su come rendere Internet migliore in futuro.</p> <p>Argomenti del corso</p> <ul style="list-style-type: none">-Principi fondamentali: architettura end-to-end, nomi e indirizzi, segnalazione, segmentazione, randomizzazione, indirectione, 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.
Testi di riferimento	<p>Reti di calcolatori e internet. Un approccio top-down, 6/Ed. James F. Kurose, Keith W. Ross, Pearson Education Italia, 2013</p> <p>Communication Networks An Optimization, Control and Stochastic Networks Perspective, R. Srikant, Lei Ying, Cambridge University Press, 2014</p>
Obiettivi formativi	<p>Gli obiettivi del corso sono:</p> <ul style="list-style-type: none">- 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.
Prerequisiti	Reti di Elaboratori
Modalità di verifica dell'apprendimento	Esame scritto e presentazione orale
Frequenza	Facoltativa



Testi in inglese

Tipo testo	Testo
Lingua insegnamento	Italian
Contenuti	<p>In the past two decades, the Internet has grown from a research tool to a fundamental part of society; something we all take for granted and use everyday. In this class we'll explore why the infrastructure was designed this way, and the underlying principles and design decisions. We'll examine the pros and cons of the current design, and give some thought to how we can make the Internet better in future.</p> <p>Topics</p> <ul style="list-style-type: none">-Fundamental principles: end-to-end design, naming and addressing, signalling, segmentation, randomization, indirection, multiplexing, virtualization, scalability.-Network resource allocation: traffic engineering, congestion control as a resource allocation problem, TCP.-Router design: input-output-queueing, packet classification, scheduling.-Content Centric Networks: content distribution networks, names vs. addresses, network caches.-BitTorrent: protocol specifications, performance modeling.-Cellular networks: architecture and principles, evolution of the standards.-Sensor networks: basic principles and main issues.-Wireless networking: complements on 802.11, Bluetooth, ad-hoc networks, opportunistic routing and scheduling.
Testi di riferimento	<p>Computer Networking: A Top-Down Approach, 6/E James F. Kurose, Keith W. Ross, Addison-Wesley, 2013</p> <p>Communication Networks An Optimization, Control and Stochastic Networks Perspective, R. Srikant, Lei Ying, Cambridge University Press, 2014</p>
Obiettivi formativi	<p>The goals for this class are:</p> <ul style="list-style-type: none">- To become familiar with the state of the art in networking research: network architecture, protocols and systems.- To gain some practice in reading research papers and critically understand them.- To learn how to effectively present and discuss in public a networking topic.
Prerequisiti	Computer Networks
Modalità di verifica dell'apprendimento	Written test and oral presentation
Frequenza	Optional

Testi del Syllabus

Docente	CANCELLIERE ROSSELLA	Matricola: 013978
Anno offerta:	2014/2015	
Insegnamento:	MFN0977 - RETI NEURALI (Neural Networks)	
Corso di studio:	008515 - INFORMATICA (Computer science)	
Anno regolamento:	2013	
CFU:	6	
Settore:	INF/01	
Tipo attività:	B - Caratterizzante	
Partizione studenti:	-	
Anno corso:	2	
Periodo:	Primo Semestre	



Testi in italiano

Tipo testo	Testo
Lingua insegnamento	italiano
Contenuti	<p>Definizione di rete neurale e di neurone. Percettrone, regola delta. Rete neurale a propagazione in avanti, addestramento a retropropagazione dell'errore. Apprendimento non supervisionato: modello di Hopfield e suo uso come memoria associativa. Self-organizing maps e loro uso come modelli psicologici.</p> <p>Reti radiali. Modello ELM (Extreme Learning Machine). Singular Value Decomposition. Macchine di Boltzmann e apprendimento stocastico. Reti neurali profonde, caratteristiche e problemi aperti.</p> <p>Implementazione dei principali modelli in ambiente Matlab.</p>
Testi di riferimento	S. Haykin - Neural Networks: a Comprehensive Foundation - IEEE Press C. M. Bishop - Pattern recognition and Machine Learning - Springer eds. I. H. Witten, E. Frank - Data Mining - 2 ed. - Elsevier eds.
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.
Prerequisiti	<p>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</p> <p>Eventuali corsi propedeutici (forniscono le "competenze attese in ingresso"): Analisi Matematica, Calcolo Matriciale e Ricerca Operativa, Programmazione, Sistemi intelligenti.</p>
Modalità di verifica dell'apprendimento	L'esame è orale, con verifica delle conoscenze acquisite sia attraverso domande che con la discussione degli algoritmi implementati.
Risultati di 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 in vari ambiti come ad esempio nell'apprendimento automatico e nel data mining.



Testi in inglese

Tipo testo	Testo
Lingua insegnamento	italian
Contenuti	Definition of neural network and of neuron. Perceptron, Delta rule. Feedforward neural networks, backpropagation training. Non supervised training: use of Hopfield model as associative memory. Self organizing maps and their use as psychological models. Radial basis functions neural networks. ELM training. Singular Value Decomposition. Boltzmann machines and stochastic training. Deep neural networks main features, and open problems. Main models realization in Matlab environment.
Testi di riferimento	S. Haykin - Neural Networks: a Comprehensive Foundation - IEEE Press C. M. Bishop - Pattern recognition and Machine Learning - Springer eds. I. H. Witten, E. Frank - Data Mining - 2 ed. - Elsevier eds.
Obiettivi formativi	Acquisition of the basic knowledge on the main neural networks models and their applications
Modalità di verifica dell'apprendimento	Oral examination
Risultati di apprendimento attesi	Critical knowledge and use of the main neural networks algorithms

Testi del Syllabus

Docente	BERGADANO FRANCESCO	Matricola: 013238
Anno offerta:	2014/2015	
Insegnamento:	MFN0952 - SICUREZZA II (Computer and Network Security II)	
Corso di studio:	008515 - INFORMATICA (Computer science)	
Anno regolamento:	2014	
CFU:	6	
Settore:	INF/01	
Tipo attività:	B - Caratterizzante	
Partizione studenti:	-	
Anno corso:	1	
Periodo:	Primo Semestre	



Testi in italiano

Tipo testo	Testo
Lingua insegnamento	Italian
Contenuti	Gestione dell'Identità digitale e sistemi di controllo di accesso
Testi di riferimento	Materiale disponibile su Moodle
Obiettivi formativi	Fornire competenza ed esperienza pratica sul tema del corso
Prerequisiti	Sicurezza I
Propedeutico a	nessun corso successivo
Modalità di verifica dell'apprendimento	Discussione in laboratorio ed esame orale
Frequenza	Necessaria



Testi in inglese

Tipo testo	Testo
Lingua insegnamento	Italian
Contenuti	Identity and Access Management
Testi di riferimento	Slides and documents available on Moodle
Obiettivi formativi	Provide competence and hand-on experience on the course's topics
Prerequisiti	Security I
Propedeutico a	no further course
Modalità di verifica dell'apprendimento	Discussion during lab and oral exam
Frequenza	Necessary

Testi del Syllabus

Docente

BOELLA GUIDO

Matricola: **014004**

Anno offerta:

2014/2015

Insegnamento:

MFN0949 - SISTEMI COGNITIVI (Cognitive Systems)

Corso di studio:

008515 - INFORMATICA (Computer science)

Anno regolamento:

2014

CFU:

9

Settore:

INF/01

Tipo attività:

B - Caratterizzante

Partizione studenti:

-

Periodo:

Secondo Semestre



Testi in italiano

Tipo testo

Testo

Lingua insegnamento

Italiano

Contenuti

INTRODUZIONE ALLA SINTASSI COMPUTAZIONALE

- Linguistica Computazionale Generale
- Introduzione allo studio formale del linguaggio
- Morfologia, Sintassi, Semantica formale
- Parsing e Generazione
- Traduzione Automatica

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. Questo modulo prevede un numero variabile di esercitazioni, durante le quali saranno proposti problemi legati agli argomenti trattati a lezione.

- Introduzione alle ontologie formali
- 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

Testi di riferimento

INTRODUZIONE ALLE ONTOLOGIE FORMALI E ALLA SEMANTICA LESSICALE

SPEECH and LANGUAGE PROCESSING, Daniel Jurafsky and James H. Martin, "An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition", Second Edition, Prentice Hall, 2009. Capp. 19, 20, 23.1.

Durante il corso sarà inoltre fornito un elenco di articoli di approfondimento.

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.

Prerequisiti

Competenze attese in ingresso (richieste all'inizio del corso).

Conoscenze di base di algoritmica e logica formale.

Si richiedono nozioni di base di Intelligenza Artificiale, in particolare relative all'area di Rappresentazione della Conoscenza.

Eventuali corsi propedeutici, che forniscono le "competenze attese in ingresso": Sistemi Intelligenti (propedeuticità consigliata).

Propedeutico a

--

Note

nessuna

Tipo testo

Testo

Modalità di verifica dell'apprendimento

INTRODUZIONE ALLA SINTASSI COMPUTAZIONALE

Progetto di programmazione su temi di trattamento automatico del linguaggio naturale

INTRODUZIONE ALLE ONTOLOGIE FORMALI E ALLA SEMANTICA LESSICALE

L'esame consiste nella discussione delle esercitazioni, nell'illustrazione delle risorse linguistiche utilizzate, e nella discussione di un articolo a scelta fra quelli proposti all'inizio del corso (concordato con il docente).

Risultati di apprendimento attesi

Competenze attese in uscita (acquisite durante il corso).

- Conoscenza delle basi della Scienza Cognitiva;
- Conoscenza delle principali risorse per l'analisi della semantica lessicale e delle ontologie formali;
- Conoscenza di alcune tecniche di elaborazione del linguaggio naturale.

Frequenza

La frequenza è consigliata, anche se non obbligatoria.



Testi in inglese

Tipo testo	Testo
Lingua insegnamento	Italian
Contenuti	<p>INTRODUCTION TO COMPUTATIONAL SYNTAX</p> <ul style="list-style-type: none">- Introduction to Computational linguistics- Formal approach to natural language- Formal morphology, syntax, semantics- Parsing and Generation- Automatic translation <p>INTRODUCTION TO FORMAL ONTOLOGIES AND TO LEXICAL SEMANTICS</p> <p>Introduction to formal ontologies, and their usage in computational linguistics and Semantic Web. Such module includes programming exercises (and homework assignments, as well) to get familiar with technologies and theories.</p> <ul style="list-style-type: none">- introduction to formal ontologies- Knowledge representation and structured representation systems- The KL-ONE system- The WordNet system- The BabelNet system- FrameNet- Languages for data modeling: XML, RDF and SPARQL- OWL- ConceptNet- Ontology development with Protegé
Testi di riferimento	<p>INTRODUCTION TO FORMAL ONTOLOGIES AND TO LEXICAL SEMANTICS</p> <p>SPEECH and LANGUAGE PROCESSING, 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.</p> <p>Further readings (research articles and book chapters) from the literature will be made available on the course site.</p>
Obiettivi formativi	The course will provide basic knowledge on cognitive systems, an introduction on formal ontologies and on lexical semantics in the frame of Natural Language Processing.
Prerequisiti	<p>Basic knowledge on algorithms and formal logics.</p> <p>Introductory level notions in Artificial Intelligence, with special focus on Knowledge Representation are also useful to fully attend the course.</p> <p>Having taken the course "Intelligent Systems" is recommended.</p>
Propedeutico a	--
Note	none
Modalità di verifica dell'apprendimento	<p>INTRODUCTION TO COMPUTATIONAL SYNTAX</p> <p>Programming project on NLP.</p>

Tipo testo

Testo

INTRODUCTION TO FORMAL ONTOLOGIES AND TO LEXICAL SEMANTICS

The exam will be a discussion starting from the exercises and homework assignments, including the resources (and underlying approaches) used therein, and the presentation of a research paper.

Risultati di apprendimento attesi

Basic knowledge of computational Linguistics:
- introductory view on the bases of Cognitive Science;
- knowledge on the main resources for the analysis of lexical semantics and formal ontologies;
- knowledge of some basic techniques for the natural language processing.

Frequenza

Attending the lessons is strongly recommended, although not strictly required.

Testi del Syllabus

Docente	LUCENTEFORTE MAURIZIO	Matricola: 014046
Anno offerta:	2014/2015	
Insegnamento:	MFN0978 - SISTEMI DI REALTA' VIRTUALE (Virtual Reality Systems)	
Corso di studio:	008515 - INFORMATICA (Computer science)	
Anno regolamento:	2013	
CFU:	9	
Settore:	INF/01	
Tipo attività:	B - Caratterizzante	
Partizione studenti:	-	
Anno corso:	2	
Periodo:	Primo Semestre	



Testi in italiano

Tipo testo

Testo

Lingua insegnamento

Italiano

Contenuti

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

Tipo testo

Testo

-Stereo e surround
-HRTF (Head-Related Transfer Function)

Laboratorio di Realtà Virtuale
-Introduzione allo sviluppo di applicazioni 3D realtime con Unity3D.

Testi di riferimento

-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.

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.

Prerequisiti

Buona conoscenza delle tecniche di analisi e di calcolo matriciale

Propedeutico a

Preparazione della prova finale in una visione interdisciplinare.

Modalità di verifica dell'apprendimento

La prova d'esame comprende un progetto di laboratorio ed un colloquio orale

Risultati di apprendimento attesi

Acquisizione delle conoscenze fondamentali per la navigazione di ambienti di sintesi mediante l'utilizzo di motori di realtà virtuale.



Testi in inglese

Tipo testo

Testo

Lingua insegnamento

Italian

Contenuti

Virtual reality

- An overview
- Definitions, problems and solutions
- Applications

The Graphics Rendering Pipeline

- Architecture
- The Application Stage
- The Geometry Stage
- The Rasterizer Stage

Mathematics for virtual environments

- Points and vectors
- Operations with vectors
- Equation of the plane
- Directions and angles
- Transformation matrix
- Quaternions: definitions and operations

Visual Appearance

- Light Sources
- Materials
- Shading

Global Illumination

- Shadows
- Ray Tracing
- Radiosity

Construction of a 3D scene

- View frustum
- Culling
- Clipping
- Removing or hidden faces: object space and image space approaches

Recovery of the third dimension

- Prospective projection
- Reconstruction of the shapes from their shadows
- Stereovision
- Calculus of conjugate points
- Outline of 3D formats and compression techniques

Simulation-Animation

- Joints: direct kinematics, inverse kinematics, dynamics
- Rigid bodies: elastic and inelastic collision, static and dynamic friction
- Natural phenomena

Intersections and Collisions

- Basic analytical geometry
- Bounding-box
- Intersection of line-plane, line-sphere, line-polygon, line-polyhedron, polyhedron-polyhedron
- Intersection of bounding boxes
- Pruning techniques: bounding volume, space partition, hierarchical bounding volume.

3D audio

- Stereo and Surround

Tipo testo

Testo

-HRTF (Head-Related Transfer Function)

Virtual Reality Laboratory

-Introduction to realtime 3D application development with Unity3D.

Testi di riferimento

- 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.

Obiettivi formativi

The fundamental goals of this class are to furnish homogeneous and interdisciplinary knowledge that concurs to the creation of virtual reality applications along with the instruments that permit to provide real-time rendering.

Prerequisiti

Good knowledge of mathematics and matrix calculus.

Propedeutico a

Preparation for the final exam with an interdisciplinary attitude.

Modalità di verifica dell'apprendimento

The exam consists in a laboratory project followed by oral examination.

Risultati di apprendimento attesi

By the end of the course the student will acquire the fundamental elements for the design of simulated environments using virtual reality engines.

Testi del Syllabus

Docente	PETRONE GIOVANNA	Matricola: 014322
Anno offerta:	2014/2015	
Insegnamento:	MFN0941 - SVILUPPO SOFTWARE PER COMPONENTI E SERVIZI WEB (Component Software and Web Services Development)	
Corso di studio:	008515 - INFORMATICA (Computer science)	
Anno regolamento:	2014	
CFU:	9	
Settore:	INF/01	
Tipo attività:	B - Caratterizzante	
Partizione studenti:	-	
Anno corso:	1	
Periodo:	Primo Semestre	



Testi in italiano

Tipo testo

Testo

Lingua insegnamento

Italiano

Contenuti

Introduzione alle 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.

Architetture per la progettazione di applicazioni client/server distribuite sulla rete dalle soluzioni two-tiers alle three-tiers e tipologie varie di middleware e di oggetti distribuiti: i business objects distribuzione dei processi applicativi nelle reti Internet ed Intranet. Architettura per oggetti distribuiti dello standard CORBA e ORB, gli Object Request Broker

L'architettura Java 2 Enterprise
Enterprise Java Beans
le interfacce utente:
Java Server Pages
JavaServer Pages Standard Tag Library
JavaServer Faces
XML per l'interscambio di dati.
Web Services
Standards: WSDL, SOAP, UDDI
Orchestrazione: BPEL
Coreografia: WS-CDL

Cenni di soluzioni per:
Enterprise Application Integration
Service Oriented Architecture.
L'utilizzo di UML (Unified Modeling Language) nel progetto preliminare di un'applicazione distribuita
Definizione e successiva realizzazione dell'applicazione distribuita in Java ed EJB, utilizzando l'application server di Sun J2EE 1.5 SDK e Netbeans ed Eclipse, l'ambiente di sviluppo Sun per J2EE.

Introduzione all'Xtreme programming per OO analysis e design
Cenni:
Project management
Managing for Excellence
Testing
Controllo di versione:
In lab per il progetto utilizzo di Subversion o GIT

Strumenti: Github, Bootstrap
In Lab Applicazione Android e utilizzo di Open API : Google, Facebook, Twitter -NEW
Deploy di app su una Cloud Platform: GAE e/o Heroku

Testi di riferimento

G. Alonso and F. Casati and H. Kuno and V. Machiraju, Web services - Concepts, architectures and applications, Springer 2004

Orfali, Harkey, Client/Server Programming with Java and CORBA, John Wiley & sons

Budi Kurniawan: "Java for the Web with Servlets, JSP, and EJB: A Developer's Guide to J2EE Solutions", Paperback

Tipo testo

Testo

Obiettivi formativi

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. * Permettere agli studenti di familiarizzare con terminologie di uso comune nell'industria moderna quali: sviluppo per componenti, architetture multilivello, middleware. * Realizzare in laboratorio 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.

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.

Prerequisiti

Laboratorio di Servizi Web, Tecnologie Web

Propedeutico a

Nessun corso.

Modalità di verifica dell'apprendimento

L'esame consiste:

1. nella presentazione e discussione della realizzazione di una applicazione three-tier, distribuita in rete, scritta in Java con l'utilizzo di EJB, Java Server Pages, XML o HTML, documentata in UML
2. di un breve colloquio orale sugli argomenti trattati a lezione.

La realizzazione dell'applicazione può essere fatta da un gruppo di 3-4 studenti che vengono seguiti su appuntamento per e-mail e singolarmente testati nella presentazione e dimostrazione del funzionamento dell'applicazione.

Risultati di apprendimento attesi

Capacità di costruzione di un sistema Web di medie dimensioni e di gestione di tale progetto sia dal punto di vista del Project Management che del completo ciclo di vita del software

Frequenza

NON obbligatoria, ma consigliata per progetto di laboratorio



Testi in inglese

Tipo testo

Testo

Lingua insegnamento

Italian

Contenuti

Overview of software architectures from distributed Client-Server, two-tiers and three-tiers architectures.

Introduction to various architectures suitable to build distributed client/server applications, two-tiers and three-tiers, various types of middleware and distributed objects: business objects. CORBA, Object Request Broker.

Java 2 Enterprise Architecture
Enterprise Java Beans
User interfaces:
Java Server Pages
JavaServer Pages Standard Tag Library
JavaServer Faces
XML for data exchange.
Web Services
Standards: WSDL, SOAP, UDDI
Orchestration: BPEL
Choreography: WS-CDL

Solutions overview for:
Enterprise Application Integration
Service Oriented Architecture.
Exploitation of UML (Unified Modeling Language) to design a distributed application.
Design and implementation of a Java distributed application using EJBs , running on an application server. Netbeans and Eclipse will be used.

Introduction to Xtreme programming for OO analysis and design
Briefly:
Project management
Managing for Excellence
Testing
Version control Subversion or GIT

Environments and tools: GitHub, Bootstrap
Applicazione Android and usage of Open API : Google, Facebook, Twitter
Deploy of an app on a Cloud Platform: GAE e/o Heroku

Testi di riferimento

G. Alonso and F. Casati and H. Kuno and V. Machiraju, Web services - Concepts, architectures and applications, Springer 2004

Orfali, Harkey, Client/Server Programming with Java and CORBA, John Wiley & sons

Budi Kurniawan: "Java for the Web with Servlets, JSP, and EJB: A Developer's Guide to J2EE Solutions", Paperback

Obiettivi formativi

Provide students with an overview of technologies used in developing mission critical systems, with a particular focus on industrial solutions for e-business applications and Cloud Computing solutions.

Methodologies of Project Management will be presented in order to manage software projects starting from requirements elicitation to testing. The Project Management basis together with the exploitation of the Agile Software Engineering methodology will be applied during the development of the project work.

Students will familiarize with terminology used in modern software

Tipo testo

Testo

industries such as component based development, multi-tiers architecture, middleware, Java Enterprise environment.

The project will involve analysis and design of the application, using UML and exploiting J2EE environment and Android development tools. Moreover, a particular attention will be given to Social Network integration, as an example of heterogeneous services integration.

Design and implementation of a distributed J2EE application, running on Glassfish and developing using Netbeans and Eclipse, IDE for J2EE development. Web Services SOAP and RESTful will be implemented.

Prerequisiti

Laboratorio di Servizi Web, Tecnologie Web

Propedeutico a

No course.

Modalità di verifica dell'apprendimento

The exam consist of:

1. a presentation of a designed and developed distributed three-tier application, written in JAVa exploiting EJB, Java Server Pages, XML o HTML, documented with UML
2. a brief oral test on the subjects not included in the project

The project can be developed in a group of 3-4 students. Part of the project will be executed during laboratories hours and students will can discuss regularly by appontment.

Frequenza

It is not mandatory, but highly recommended for better learn from the project work.

Testi del Syllabus

Docente	GRANGETTO MARCO	Matricola: 014469
Anno offerta:	2014/2015	
Insegnamento:	MFN0961 - TEORIA DELL'INFORMAZIONE (Information theory)	
Corso di studio:	008515 - INFORMATICA (Computer science)	
Anno regolamento:	2014	
CFU:	9	
Settore:	INF/01	
Tipo attività:	B - Caratterizzante	
Partizione studenti:	-	
Anno corso:	1	
Periodo:	Secondo Semestre	



Testi in italiano

Tipo testo

Testo

Lingua insegnamento

Italiano

Contenuti

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).

Testi di riferimento

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.

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.

Prerequisiti

L'organizzazione del modulo presuppone una buona conoscenza delle nozioni fornite dai corsi di probabilità e statistica e più in generale dai corsi di matematica.

Propedeutico a

Comprensione di elementi fondamentali delle reti di comunicazione e rappresentazione dati multimediali.

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.

Risultati di 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.

Testi in inglese

Tipo testo	Testo
Lingua insegnamento	Italian
Contenuti	<p>The course is structured into three parts.</p> <p>The first part deals with classical information theory. It includes the definition of information and source models, the concept of entropy, source coding, the first Shannon theorem (the source coding theorem), unique decodeability, Huffman codes and their optimality, noisy channel models, definition of the channel capacity, second Shannon theorem (the noisy channel theorem).</p> <p>The second part is devoted to the study of practical source and channel coding schemes, employed in several applications and communication systems. Among the source coding techniques arithmetic codes, di Lempel-Ziv-Welch coding and some image and video coding standards will be analyzed. Channel coding schemes based on linear block codes, cyclic redundancy check and convolutional codes will be studied.</p> <p>In the last part of the course a set of advanced topics will be presented and studied in depth, including some experimental activities as well. The addressed topics will be selected according to their relevance in the research field or for their widespread diffusion in commercial systems</p>
Testi di riferimento	<p>Thomas M. Cover, Joy A. Thomas, Elements of Information Theory, Wiley, 1991 , 2006(2nd edition).</p> <p>David J. C. MacKay, Information Theory, Inference and Learning Algorithms, Cambridge University Press, 2003.</p> <p>Robert G. Gallager, Information Theory and Reliable Communication (Paperback), Wiley, 1968.</p>
Obiettivi formativi	<p>The course provides essential elements of information theory allowing one to understand, analyze and design information coding and transmission systems. The acquired competences will be the knowledge of the Shannon theory and of its many applications in computer science fields.</p>
Prerequisiti	<p>Good knowledge of mathematics, probability and statistics is required.</p>
Propedeutico a	<p>Acquisition of fundamental elements of telecommunication networks and representation of multimedia data.</p>
Modalità di verifica dell'apprendimento	<p>The final exam is based on written test and oral examination.</p>
Risultati di apprendimento attesi	<p>By the end of the course the student will have acquired the fundamental concepts of information theory and will be able to apply the results to the analysis and design of information coding and transmission systems.</p>

Testi del Syllabus

Docente	DI LEVA ANTONIO	Matricola: 010913
Anno offerta:	2014/2015	
Insegnamento:	MFN0948 - TRATTAMENTO DEI DATI E DEI PROCESSI AZIENDALI (Data and Process Management in Enterprises)	
Corso di studio:	008515 - INFORMATICA (Computer science)	
Anno regolamento:	2014	
CFU:	9	
Settore:	INF/01	
Tipo attività:	B - Caratterizzante	
Partizione studenti:	-	
Periodo:	Secondo Semestre	



Testi in italiano

Tipo testo

Testo

Contenuti

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.

Testi di riferimento

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)

Obiettivi formativi

Fornire allo studente competenze generali e pratiche di base sull'analisi organizzativa e funzionale delle aziende e sulla ingegnerizzazione e reingegnerizzazione 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 che dalla raccolta e analisi dei requisiti e dall'analisi dell'organizzazione conducono al progetto dei nuovi processi ottimizzati dell'azienda. Infine una parte del corso, circa 30 ore, verrà dedicata ad un argomento monografico che potrà cambiare in ogni anno accademico; per l'anno in corso tale argomento riguarderà la valutazione dei processi mediante dei Key Performance Indicators (KPI) - indicatori di efficienza da usare nella costruzione di cruscotti per la gestione aziendale.

Testi del Syllabus

Docente	DI LEVA ANTONIO	Matricola: 010913
Anno offerta:	2014/2015	
Insegnamento:	MFN0948 - TRATTAMENTO DEI DATI E DEI PROCESSI AZIENDALI (Data and Process Management in Enterprises)	
Corso di studio:	008515 - INFORMATICA (Computer science)	
Anno regolamento:	2013	
CFU:	9	
Settore:	INF/01	
Tipo attività:	B - Caratterizzante	
Partizione studenti:	-	
Anno corso:	2	
Periodo:	Secondo Semestre	



Testi in italiano

Tipo testo

Testo

Contenuti

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.

Testi di riferimento

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)

Obiettivi formativi

Fornire allo studente competenze generali e pratiche di base sull'analisi organizzativa e funzionale delle aziende e sulla ingegnerizzazione e reingegnerizzazione 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 che dalla raccolta e analisi dei requisiti e dall'analisi dell'organizzazione conducono al progetto dei nuovi processi ottimizzati dell'azienda. Infine una parte del corso, circa 30 ore, verrà dedicata ad un argomento monografico che potrà cambiare in ogni anno accademico; per l'anno in corso tale argomento riguarderà la valutazione dei processi mediante dei Key Performance Indicators (KPI) - indicatori di efficienza da usare nella costruzione di cruscotti per la gestione aziendale.

Testi del Syllabus

Docente	BALBO GIANFRANCO	Matricola: 011347
Anno offerta:	2014/2015	
Insegnamento:	MFN0899 - VALUTAZIONE DELLE PRESTAZIONI: SIMULAZIONE E MODELLI (SIMULATION AND MODELLING)	
Corso di studio:	008515 - INFORMATICA (Computer science)	
Anno regolamento:	2014	
CFU:	9	
Settore:	INF/01	
Tipo attività:	B - Caratterizzante	
Partizione studenti:	-	
Anno corso:	1	
Periodo:	Secondo Semestre	



Testi in italiano

Tipo testo

Testo

Contenuti

I modelli discussi in questo corso sono delle rappresentazioni probabilistiche in cui vari aspetti della realta' vengono espressi sotto forma di reti di stazioni di servizio in fronte alle quali e' 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 (nei casi piu' semplici) e di tecniche simulative in quelli piu' complessi. 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 modalita' 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 e' 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.

Testi di riferimento

Leonard Kleinrock, "Sistemi a Coda: Introduzione alla Teoria delle Code", Hoepli, 1992.

Giuseppe Iazeolla, "Introduzione alla Simulazione Discreta", Boringhieri, 1980.

Sheldon M. Ross, "A Course in Simulation", Macmillan, 1990.

Obiettivi formativi

Modellazione ed analisi di sistemi di traffico con particolare riferimento alle reti di telecomunicazione ed ai sistemi di calcolo distribuiti.

Prerequisiti

Conoscenze di base di Analisi matematica, Probabilita' e Statistica, Architetture dei Calcolatori, Sistemi Operativi

Modalità di verifica dell'apprendimento

Sviluppo di un progetto assegnato dal docente e discussione orale degli aspetti teorici dei metodi utilizzati per la soluzione dei modelli analizzati durante il progetto

Risultati di apprendimento attesi

Alla fine del corso gli studenti conosceranno le tecniche di base utilizzate nella pratica per studiare i modelli di traffico caratterizzati dalla presenza di punti di congestione (code) e sapranno orientarsi per scegliere la tecnica piu' adatta ad affrontare problemi reali



Testi in inglese

Tipo testo

Testo

Contenuti

Models discussed in this course correspond to probabilistic representations in which different aspects of reality are expressed in form of networks of servers in front of which queues may develop because of the existence of congestion and/or synchronization phenomena.

Operational Analysis is discussed in the first part of the course to introduce the basic concepts of this subject and to tackle the modelling and analysis of relatively simple case studies. When the probabilistic characteristics of the models become more complex, the course introduces Markov chains and discusses how this mathematical framework can be used to study in a nice and elegant manner intrinsically complex models, but with computational costs that may be extremely large.

When Markov chains turn out to be inadequate for tackling the complexity of real cases, Discrete Event Simulation becomes the most suitable analysis technique.

The simulation of a probabilistic models consists of writing a program capable of reproducing (with a certain level of abstraction) the behaviour of the model. The execution of the simulation program corresponds to the evolution of the model, starting from an initial state and ending in a predefined final state. The execution of the program is driven by random number generators and thus corresponds to one of the possible evolutions of the model. Measures performed during the simulation are instances of random variables; the statistical analysis of the simulation output is thus conducted to obtain interval estimates (confidence intervals) of the performance indices of the models.

Once the basic elements of modelling and analysis of these models have been discussed, the course focuses on a few reference applications coming from telecommunication and distributed computing contexts and shows how components of systems of this type can be modelled and evaluated for the assessment of their performance.

Obiettivi formativi

Modelling and analysis of traffic systems with a particular emphasis on telecommunication networks and distributed computer systems

Prerequisiti

Basic knowledge of Calculus, Probability and Statistics, Computer Architecture, Operating Systems

Modalità di verifica dell'apprendimento

Implementation and analysis of a project suggested by the instructor and oral discussion of the theoretical aspects of the techniques used to solve the models studied during the development of the project

Risultati di apprendimento attesi

Students attending this course are expected to achieve a good knowledge of the basic techniques used in practice to study traffic models characterized by the presence of congestion points (queues) and will have the capability of choosing the method that is most suitable to study real problems

Testi del Syllabus

Docente	DONATELLI SUSANNA	Matricola: 013239
Anno offerta:	2014/2015	
Insegnamento:	MFN0959 - VERIFICA DEI PROGRAMMI CONCORRENTI (Software Reliability Methods)	
Corso di studio:	008515 - INFORMATICA (Computer science)	
Anno regolamento:	2014	
CFU:	9	
Settore:	INF/01	
Tipo attività:	B - Caratterizzante	
Partizione studenti:	-	
Periodo:	Secondo Semestre	



Testi in italiano

Tipo testo	Testo
Lingua insegnamento	italiano
Contenuti	<p>Introduzione alla verifica dei sistemi Un primo linguaggio di specifica, le Reti di Petri: definizione, proprietà e tool disponibili Un secondo linguaggio di specifica, le algebre dei processi: definizione, proprietà e tool disponibili Esprimere proprietà: le logiche temporali lineari e branching e i relativi tool Esprimere il tempo: gli automi temporizzati e le logiche branching temporizzate</p> <p>Il corso "Parte A" da 6 cfu non affronta l'ultimo argomento relativo all'espressione del tempo</p>
Testi di riferimento	<ol style="list-style-type: none">1.Christel Baier and Jost-Pieter Katoen, Model Checking - MIT Press2. Doron Peled, Software Reliability Models,
Obiettivi formativi	Obiettivo del corso e' 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.
Prerequisiti	Consigliato il corso di MCAD (modelli concorrenti e algoritmi distribuiti)
Propedeutico a	Nessun corso particolare
Modalità di verifica dell'apprendimento	Il corso ha due modalità di esame, a scelta dello studente: Tipo 1: esame scritto 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
Risultati di 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. 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 riceverà 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"
Frequenza	Consigliata



Testi in inglese

Tipo testo	Testo
Lingua insegnamento	italian
Contenuti	<p>This is a course on system verification, with special emphasis on software verification and with a blend of theoretical and applicative issues, all supported by first-hand-experience on "almost real" verification tools.</p> <p>Topics covered by the course are:</p> <ul style="list-style-type: none">- an introduction to system verification- a first specification language: Petri nets; definition, properties and available tools- a second specification language: process algebras;- properties' specification languages: linear temporal logic, branching temporal logics and available model-checking tools- expressing timed models and timed properties: timed automata and timed branching logics <p>The 6 cfu version of the course does not cover the "timed" part of the material</p>
Testi di riferimento	<ol style="list-style-type: none">1.Christel Baier and Jost-Pieter Katoen, Model Checking – MIT Press2. Doron Peled, Software Reliability Models,
Obiettivi formativi	The course aims at providing to student the theoretical and the practical background necessary to be able to verify systems, with a special emphasis on software systems.
Prerequisiti	Basic knowledge of concurrency theory, as acquired in the MCAD course (modelli concorrenti e algoritmi distribuiti)
Propedeutico a	No specific course
Modalità di verifica dell'apprendimento	<p>There are two different types of exams for this course, and the student can choose one of the two.</p> <p>Type1. Written exam on basic course topics and an extensive set of lab assignments to be discussed with the teacher.</p> <p>Type2. Oral exam on the whole set of course topics and a limited set of lab assignment to be discussed with the teacher</p>
Risultati di apprendimento attesi	<p>At the end of the course the student will be able to specify concurrent systems/programs in a formal specification language and to use software tools to define and check system properties.</p> <p>Typical properties of interest include deadlock freedom, fairness and liveness, but also user-specified properties expressed in various forms of temporal logics. Verification algorithms include model-checking of at least three different logics.</p> <p>Examples of typical temporal logic properties of interest for this course are: "if process P sends a message, then no further message would be received until an acknowledge is received" or "if P sends a message, then P will receive an ack within 5 time units".</p>
Frequenza	Suggested, but not mandatory