

REVISIONE 1.1 – 26/11/2014: acquisita certificazione grin, e di conseguenza la guida ora riporta in prima pagina il bollino della certificazione GRIN + modifica descrizione Biblioteca

REVISIONE 1.2 – 12/10/2016: modifiche sugli insegnamenti "Tecnologie web" e "Servizi web" e inserimento codice per Prolungamento Stage come da delibera CCL-LM di 19-02-2016

GUIDA AL CORSO DI LAUREA TRIENNALE IN INFORMATICA (d.m. 270) E MANIFESTO DEGLI STUDI

Corso di laurea triennale 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



Prefazione

Gentile lettrice, gentile lettore,

questa è la guida del corso di Laurea Triennale in Informatica (secondo il DM 270, classe di laurea L31), ed è organizzata in due parti, come da indicazioni ministeriali. La prima parte è la descrizione del corso di Laurea Triennale in Informatica (secondo il DM 270, classe di laurea L31), questa descrizione costituisce il “Manifesto degli studi - Piano dell’Offerta Formativa sui tre anni per la coorte 2014”, mentre la seconda parte descrive tutti gli insegnamenti offerti per il 2014/2015 (per tutte le coorti attive), elencando per ogni insegnamento i docenti di riferimento, il programma di esame, le modalità di esame ed altre informazioni utili. La prima parte è quindi di interesse precipuamente per gli studenti che si immatricolano quest’anno (coorte 2014), che trovano nel Manifesto la descrizione del loro percorso triennale, mentre la seconda parte è di interesse anche per gli studenti delle due coorti precedenti (2012 e 2013) che devono seguire gli insegnamenti del terzo e secondo anno, rispettivamente. Ricordiamo che per gli studenti immatricolati negli anni precedenti vale il manifesto degli studi della loro coorte, già pubblicato negli anni precedenti.

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

1. La durata della Laurea triennale è di tre anni accademici. I primi due anni sono uguali per tutti gli studenti: gli insegnamenti dei primi due anni sono suddivisi fra corso A e corso B e i principali corsi di laboratorio sono poi ulteriormente suddivisi, per permettere un adeguato rapporto docente/studenti. Il terzo anno è invece articolato in tre diversi indirizzi, al fine di permettere allo studente una maggiore personalizzazione degli studi.

2. La Laurea in Informatica triennale dà accesso alla laurea Magistrale in Informatica, della durata di due anni accademici, laurea che, al momento, è anch'essa strutturata in tre indirizzi.
3. La Laurea magistrale dà accesso, previo esame di selezione, al Dottorato di Ricerca in Informatica, della durata di tre anni.
4. La Laurea triennale dà anche accesso ai corsi di master di primo livello, normalmente della durata di un anno. I master hanno un obiettivo più spiccatamente professionalizzante e vengono organizzati in collaborazione con aziende del settore e altri enti formativi, e la loro offerta varia di anno in anno.
5. Il termine "Corso di Studi" è usato in questa Guida per riferirsi al Corso di Laurea triennale in Informatica.
6. Dal 2012/2013 Università di Torino ha adottato una procedura completamente telematica per l'iscrizione agli esami e la registrazione dei voti conseguiti (con conseguente abolizione del "libretto degli esami" in forma cartacea).
7. Tutte le informazioni non riportate in questa guida (come i programmi dettagliati dei corsi, gli orari, e la localizzazione delle aule) sono reperibili dal sito Internet del Corso di Studi, all'indirizzo <http://www.informatica.unito.it>
8. Per tutto quanto riguarda la procedura di immatricolazione, pagamento tasse, supporto agli studi, assegnazione login di Ateneo per l'accesso alle procedure on-line di iscrizione, e molto altro, potete far riferimento alla [pagina principale](#) del sito di Ateneo, selezionando poi il profilo "[futuro studente](#)".
9. La struttura di questa guida segue le indicazioni Ministeriali, una diversa descrizione del Corso di Studi si può trovare sul sito del Ministero, accesso diretto [al nostro Corso di Studi](#), oppure all'intera [offerta formativa ministeriale](#).

In ultimo, una raccomandazione ed un consiglio per gli immatricolandi: è molto importante che gli studenti che si iscrivono *a tempo pieno* seguano regolarmente le lezioni e diano gli esami con regolarità, al fine di laurearsi entro i tempi previsti. La struttura degli insegnamenti e delle prove di verifica è pensata per chi segue gli studi regolarmente, facendo gli esami a tempo debito e laureandosi nel tempo previsto. Seguire i corsi e studiare con regolarità *sin dal primo giorno*, anche se non ci sono più verifiche e interrogazioni, programmate o meno, come nella scuola secondaria di II grado, renderà il vostro percorso in questo Corso di Studi più semplice, più interessante e più coinvolgente. Gli studenti iscritti *a tempo parziale* possono rivolgersi al proprio tutor (vedi in seguito) per mettere invece a punto un percorso personalizzato.

Per ulteriori informazioni, siete invitati a partecipare alla presentazione del Corso di Studi che si terrà il giorno **24 settembre 2014**, alle ore 14.00 – aula A, presso il Dipartimento di Informatica (v. Pessinetto, 12 - Torino). La presentazione è aperta anche agli studenti non ancora immatricolati.

Infine, un augurio: spero che gli anni che passerete con noi, impegnati negli Studi Universitari di Informatica, rappresentino un momento importante, divertente e stimolante della vostra crescita personale e professionale.

Buona lettura!

Susanna Donatelli, PhD,
Professore Ordinario di Informatica,
Presidente del Corso di Studi in
Informatica

Tel. 011 6706711
e-mail presccs@educ.di.unito.it

INDICE

Prefazione	1
Prima parte: Manifesto degli Studi	8
Obiettivi e sbocchi professionali	8
Obiettivi del Corso di Laurea	8
Sbocchi professionali.....	9
Certificazione di Qualità	9
Organizzazione della didattica	10
Introduzione	10
Requisiti di ammissione	11
Pre-corsi.....	12
Iscrizione a tempo pieno e a tempo parziale.....	12
Piano Carriera e responsabili.....	13
Periodi di lezione e esami.....	14
Iscrizione come seconda laurea, passaggio o trasferimento.	15
Riconoscimento Certificazioni di lingua inglese	15
Il tutorato	16
Date e scadenze	17
La Laurea triennale in Informatica	18
Generalità	18
Struttura dell'offerta formativa	21
Struttura dell'offerta formativa rispetto al RAD della classe.....	27
Supporto agli studi.....	29
Erasmus.....	29
Biblioteca.....	30
Supporto on-line ai corsi.....	30
Aule e laboratori.....	31
Zone studio	32
Supporto agli studenti Disabili.....	33
Supporto agli studenti con DSA (Disturbi Specifici di Apprendimento)	35

Indirizzi utili	36
Seconda Parte: Programmi e docenti dei Corsi per l'A.A. 14/15.....	38
Programmi, docenti e semestri per i corsi attivi nel 14/15.....	38
Programmi e docenti previsti per i corsi di futura attivazione:	41
Codici per corsi esterni.	42
Codici per convalide di corsi e competenze.	43
Programmi e altre informazioni per i corsi attivi nel 14/15 (syllabus dei corsi).....	44

Prima parte: Manifesto degli Studi

Obiettivi e sbocchi professionali

Obiettivi del Corso di Laurea

In questi anni il mondo delle imprese, dei servizi e della Pubblica Amministrazione continua a guardare con particolare interesse ai laureati in Informatica e la rapida evoluzione del settore crea sempre nuove opportunità di inserimento degli informatici nel mondo del lavoro.

Il corso di Studi fornisce una solida preparazione metodologica di base, coniugata con la comprensione delle più recenti tecnologie in diversi ambiti, dall'impresa al servizio pubblico e alle applicazioni individuali, sia su computer nel senso classico del termine (PC e server) che su dispositivi mobili. Obiettivo del corso di studi è di far acquisire le competenze necessarie sia per un rapido inserimento nel mondo del lavoro (nel settore delle tecnologie dell'informazione e della comunicazione) sia per permettere al laureato di seguire la rapida evoluzione tecnologica e di adattarsi in modo flessibile a realtà lavorative molto varie.

Il laureato in Informatica sarà in grado di utilizzare le conoscenze e le competenze acquisite nella progettazione, sviluppo e gestione di sistemi informatici; possiederà inoltre le capacità necessarie per affrontare ed analizzare problemi in vari contesti applicativi e per sviluppare sistemi informatici per la loro soluzione.

Oltre alla preparazione nel settore matematico e informatico, integrate dalle competenze in ambito economico e del diritto, il percorso formativo prevede l'acquisizione, da parte dello studente, delle competenze necessarie per lavorare in gruppo e per comprendere e produrre documentazione tecnica sia in Italiano che in Inglese.

Sbocchi professionali

I laureati in Informatica svolgeranno attività professionale negli ambiti della progettazione, organizzazione e gestione di sistemi informatici, sia in imprese produttrici di hardware o software nelle aree dei sistemi informatici e delle reti, sia nelle imprese che utilizzano le tecnologie dell'informazione nel loro settore specifico (banche, assicurazioni, enti pubblici,...). 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.

A norma del DPR 328, 5 giugno 2001, la Laurea triennale in Informatica dà titolo per l'ammissione all'esame di stato per la professione di Ingegnere dell'Informazione, Sez. B.

Certificazione di Qualità

Il GRIN, che è l'organizzazione nazionale dei ricercatori di Informatica, ha istituito, in intesa con AICA, 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 della verifica della buona organizzazione e del buon funzionamento del Corso stesso.

Organizzazione della didattica

Introduzione

Il ***Corso di Laurea in Informatica*** è un percorso triennale che porta al conseguimento del titolo di Dottore in Informatica; l'iscrizione richiede il possesso di un diploma di scuola secondaria di II grado. In accordo con la riforma degli studi universitari nota come "riforma 270", dal numero del decreto ministeriale che l'ha definita, si tratta di un percorso di studi a se stante, che trova però il suo naturale completamento nel ***Corso di Laurea Magistrale***, un percorso di studi di due anni che porta al conseguimento del titolo di Dottore Magistrale in Informatica, oppure nei ***Master di primo livello*** (normalmente della durata di un anno). Il percorso Magistrale è sempre attivo, mentre l'offerta dei Master, normalmente concordata con le aziende, anche in risposta a bandi regionali od europei, non ha una cadenza, una data di inizio e un ambito tematico ricorrente, e quindi le informazioni non sono riportate in questa guida. 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 ***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 o di proiettarsi verso aspetti interdisciplinari o di ricerca.

Un concetto fondamentale per comprendere l'organizzazione dei corsi di laurea è quello di ***Credito Formativo Universitario*** (CFU). 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 triennale è costituito da 180 CFU, mentre il percorso di una Laurea Magistrale è costituito da 120 CFU e un corso di Master da 60 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 una buona conoscenza delle tematiche propedeutiche per il corso, indicati nel seguito come 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 che abbia acquisito nella Scuola Secondaria di II grado una solida preparazione di base, e che ogni studente dovrà trovare, con l'esperienza, il numero di ore e le modalità di studio individuale che meglio gli permettono di acquisire le richieste competenze.

In questa guida, i termini *corso*, *insegnamento* e *unità didattica* sono usati in modo equivalente.

Requisiti di ammissione

Sono ammessi al Corso di Laurea in Informatica gli studenti in possesso di Diploma di Scuola Media Superiore o titolo ad esso equivalente. Per frequentare il Corso di Laurea in Informatica non si richiedono prerequisiti particolari, né competenze informatiche di alcun tipo, ma è indispensabile avere propensione al ragionamento logico e una buona preparazione nelle materie di base della scuola media superiore (in primo luogo la matematica). Tale preparazione di base sarà valutata mediante un *test di accertamento (TARM: Test di Accertamento dei Requisiti Minimi)*, al quale sono tenuti a partecipare i neo-iscritti. **Il test TARM è obbligatorio e non ha carattere selettivo.** Dal 2009-2010, il test è coordinato a livello nazionale per tutti i corsi di studio in informatica, e da quest'anno è coordinato a livello nazionale dall'associazione Con-Scienze.

L'obiettivo del test è quello dell'*autovalutazione*, il test permette allo studente di capire quali siano le sue eventuali lacune o i suoi punti di forza rispetto alle

materie che si troverà ad approfondire nel Corso di Studi. Per colmare tali eventuali lacune, le prime lezioni dei corrispondenti corsi saranno dedicate al ripasso degli argomenti fondamentali.

Al test occorre prenotarsi. La prenotazione può essere fatta, a partire da fine luglio, per via telematica, all'indirizzo:

<http://di.unito.it/tarm>

Il test si svolgerà nei giorni **18, 19, 23, 25 e 26 settembre 2014**. Luogo e orario del test saranno comunicati via mail qualche giorno prima delle date sopra indicate.

Pre-corsi

Sono previsti pre-corsi per Matematica e Fisica per il recupero di eventuali lacune di preparazione. I corsi sono particolarmente consigliati agli studenti che non avranno superato il TARM. I pre-corsi sono associati ai normali insegnamenti: per Matematica si svolgeranno all'interno dell'orario dei corsi del primo anno di Matematica Discreta e Logica e di Analisi Matematica, per gli argomenti di loro competenza. Per Fisica il pre-corso è invece previsto al secondo anno, all'interno dell'orario del corso di Fisica.

Iscrizione a tempo pieno e a tempo parziale

La riforma 270 ha introdotto una distinzione importante tra gli studenti universitari: quelli iscritti a tempo pieno e quelli iscritti a tempo parziale. Gli studenti "a tempo pieno" possono laurearsi nei tre anni previsti, ma devono dedicare tutte le proprie energie (lavorative) allo studio. Considerando i CFU previsti per il Corso di Studi, si può osservare che in media lo studente deve acquisire 60 crediti all'anno, che, a 25 ore per CFU, corrispondono a 1500 ore di lavoro. Su 45 settimane, sono circa 33 ore di lavoro alla settimana.

Se però uno studente intende dedicarsi solo parzialmente allo studio (se, ad esempio, ha anche un'attività lavorativa, o se deve dedicare una parte sostanziale della sua giornata a viaggiare fra casa e università), può iscriversi a tempo parziale. Lo svantaggio è che non sarà più possibile laurearsi in tre

anni; il vantaggio è che le tasse annuali sono inferiori e che lo studente non si trova costretto a seguire un ritmo di studio eccessivo, che lo porta a sentirsi sempre "indietro" rispetto agli standard. Si tratta quindi di una scelta importante, che va fatta dopo un'attenta valutazione delle due alternative. Si noti che la scelta viene effettuata ogni anno e non ci sono problemi nel cambiarla da un anno all'altro. Per informazioni sulle modalità di pagamento delle tasse universitarie, si veda <http://di.unito.it/tasse>. Allo stato attuale lo studente iscritto a tempo pieno potrà registrare un massimo di 80 CFU all'anno, mentre lo studente iscritto a tempo parziale potrà registrare un massimo di 36 CFU all'anno.

Piano Carriera e responsabili

Gli studenti sono tenuti a presentare il piano carriera (ex carico didattico) seguendo le istruzioni della pagina: <http://di.unito.it/pianocarriera> (accesso con credenziali di Ateneo): la presentazione avviene necessariamente fra il mese di novembre 2014 e il 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 I e II anno (biennio comune con esami obbligatori per tutti) e indicare una scelta per il curriculum del III anno. Qualora lo studente voglia riservarsi di fare questa scelta negli anni successivi, come di norma avviene, potrà indicare come scelta il *percorso generico*. La scelta del curriculum potrà essere modificata in modo autonomo da parte dello studente fino a che non si siano sostenuti esami del III anno, oppure anche oltre tale limite, tramite una richiesta di modifica del piano carriera all'apposita commissione.

La compilazione del Piano Carriera è condizione necessaria per poter sostenere gli esami, anche quelli obbligatori del primo anno.

Per ulteriori informazioni e per modifiche dei piani carriera, gli studenti devono far riferimento alla *Commissione passaggi, trasferimenti e piani carriera* del Corso di Studi di Informatica (compassccs@educ.di.unito.it). Per ulteriori

informazioni sui curricula gli studenti possono rivolgersi ai relativi docenti responsabili:

- Curriculum Informazione e Conoscenza: Liliana Ardissono
(liliana@di.unito.it)

- Curriculum Linguaggi e Sistemi: Viviana Bono
(bono@di.unito.it)

- Curriculum Reti e Sistemi Informatici: Giancarlo Ruffo
(ruffo@di.unito.it)

Periodi di lezione e esami

Per il Corso di Studi in Informatica l'Anno Accademico è suddiviso in due periodi didattici di 19 settimane ciascuno (*semestri*), di cui 12 settimane di lezione ed almeno 5 settimane di sospensione per permettere agli studenti di sostenere gli esami, per i quali è prevista l'iscrizione on-line obbligatoria (<http://di.unito.it/appelli>). 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 5 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 5 gli appelli dell'anno di uno stesso insegnamento. Gli studenti possono iscriversi all'esame di un insegnamento solo se sono in regola con il pagamento delle tasse, se, inoltre, l'insegnamento fa parte del loro piano carriera per l'anno in corso (o precedenti), e se lo studente ha partecipato all'attività di valutazione dell'insegnamento stesso (procedura web per la raccolta delle opinioni degli studenti sugli insegnamenti)

Sono previste tre sessioni di laurea all'anno, normalmente nei mesi di Luglio, Ottobre e Marzo/Aprile. Le date per le sessioni di laurea della coorte 2014/2015 saranno rese note entro l'inizio del terzo anno accademico della coorte.

Iscrizione come seconda laurea, passaggio o trasferimento.

Gli studenti che intendono iscriversi come seconda laurea, passaggio da altro corsi di studio o trasferimenti da altri Atenei devono farsi pre-valutare la carriera inviando l'elenco degli esami sostenuti, completo di codice corso, CFU attribuiti, settore scientifico/disciplinare e riferimento web al programma, ai seguenti docenti:

Docente	Telefono	E-mail
Daniele Gunetti	011 - 670 67 68	gunetti@di.unito.it
Giancarlo Ruffo	011 - 670 67 71	ruffo@di.unito.it

Riconoscimento Certificazioni di lingua inglese

Gli studenti possono richiedere il riconoscimento dei certificati B1-B2 (secondo il Common European Framework) per l'esame di Lingua Inglese I compilando l'apposito [modulo](#) per il riconoscimento delle APU - Attività Pre-Universitarie. Il modulo va consegnato alla [Segreteria Studenti del Polo delle Scienze della](#)

[Natura](#), in via S.Croce 6, entro il 15 giugno di ogni anno (primo giorno lavorativo successivo, in caso il 15 giugno sia un giorno festivo).

A seguito di apposita delibera, gli esami riconosciuti saranno direttamente caricati sulla carriera degli studenti e lo studente non dovrà più sostenere Lingua Inglese I – mfn0590 – 3 CFU, facente parte dei crediti obbligatori del piano carriera del I anno.

Il tutorato

Il tutorato è stato istituito con la legge 341/90 (riforma degli ordinamenti didattici universitari): si tratta di un'attività diretta a "orientare ed assistere gli studenti lungo tutto il percorso degli studi, a renderli attivamente partecipi del processo formativo, a rimuovere gli ostacoli ad una proficua frequenza dei corsi, anche attraverso iniziative rapportate alle necessità, alle attitudini e alle esigenze dei singoli" (art. 13).

Dall'anno accademico 2003-04, il Corso di Studi in Informatica ha attivato il 'servizio' di tutorato individuale. Si tratta di un'attività che si propone di aiutare gli studenti a procedere con regolarità negli studi. A tale scopo, ad ogni studente verrà assegnato un docente di riferimento (*tutor*). Lo studente verrà invitato a partecipare agli incontri di tutorato nei tempi e nei modi stabiliti dalla commissione tutorato. Dal 2014/2015 sarà anche attivo il "tutorato di materia", per il momento previsto per il solo corso di Analisi Matematica.

Per il Tutorato sono previsti due incontri durante l'anno e uno all'inizio dell'anno successivo. Gli ultimi incontri sono particolarmente utili per pianificare eventuali recuperi per gli insegnamenti sui quali si sono avute particolari difficoltà.

Il calendario degli incontri è il seguente:

I incontro: da lunedì 23/2 a lunedì 9/3 2015

II incontro: da lunedì 1/6 a lunedì 15/6 2015

III incontro: da lunedì 28/9 a lunedì 12/10 2015

Date e scadenze

Queste sono le principali scadenze a cui dovrete prestare attenzione durante il vostro primo anno. Le date inserite in questa guida sono solo un promemoria: si consiglia di consultare le scadenze anche sul [sito web dell'Ateneo](#).

Per le immatricolazioni alla laurea triennale:

Compilazione on-line: dal 1° settembre 2014 al 10 ottobre 2014

(www.unito.it)

Consegna documentazione firmata: dal 1° settembre 2014 al 10 ottobre 2014

(e fino al 20 dicembre 2013 previo pagamento di sanzione amministrativa)

Centro Immatricolazioni, Corso Regio Parco 134/A – 10154 Torino
numero verde 800 098590 dal lunedì al venerdì dalle 8.00 alle 20.00 e il sabato dalle 8.00 alle 13.00

Per i trasferimenti, passaggi, iscrizione con abbreviazione di carriera e seconda laurea:

Dal 1° settembre 2014 al 10 ottobre 2014

(e fino al 23 dicembre 2014 previo pagamento di sanzione amministrativa)

Per maggiori informazioni: <http://di.unito.it/passtrasf>

Si presenta domanda presso la Segreteria Studenti della sede da cui lo studente proviene

Per la presentazione del piano carriera: da novembre 2014 al 31 gennaio 2015.

Per l'iscrizione agli esami: la data ultima per l'iscrizione ad ogni singolo esame è pubblicata insieme alla data dell'esame stesso, nella bacheca degli esami

della pagina personale dello studente sul sito web dell'Ateneo, di norma 4/5 giorni prima della data di esame.

Per gli orari della Segreteria Studenti, si veda:
<http://di.unito.it/segreteriaStudenti>

La Laurea triennale in Informatica

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 L-31 e la corrispondente implementazione dell'offerta effettuata dall'Università di Torino, e approvata dal Ministero. Vengono infine descritti i tre indirizzi, con l'elenco degli associati insegnamenti e la loro distinzione in caratterizzanti, affini e liberi, e la loro natura obbligatoria o meno.

Generalità

Il Corso di Laurea triennale in Informatica si articola in tre diversi percorsi (*curricula*) con i primi due anni in comune, dedicati allo studio di materie propedeutiche per i corsi successivi. Nel primo e secondo anno sono presenti corsi di matematica, corsi di introduzione alla struttura dei calcolatori e alla programmazione, e complementi di economia, diritto e fisica.

Il terzo anno lo studente dovrà scegliere tra i tre curricula: ***Informazione e Conoscenza (IC)***, ***Linguaggi e Sistemi (LS)***, oppure ***Sistemi e Reti (SR)***. Tutti i percorsi affiancano alla didattica svolta in aula delle ore di didattica in

laboratorio e prevedono la formazione anche in ambiti disciplinari di cultura generale come diritto, lavoro di gruppo e conoscenza della lingua inglese. Il percorso formativo viene concluso con lo sviluppo di un progetto, di norma svolto presso un'azienda, che offre allo studente una prima esperienza lavorativa.

Come ricordato in precedenza, per il Corso di Laurea in Informatica è stabilito che ogni CFU (che, ricordiamo, corrisponde per legge a 25 ore di lavoro per lo studente) sia equivalente a 10 ore di lezione/esercitazione + 15 ore di studio individuale. Per facilitare l'inserimento nel mondo universitario, si propone un primo anno più leggero, di soli 57 CFU, seguito da due anni di 60 e 63 CFU. Gli studenti potranno comunque, secondo il *Regolamento Studenti di Ateneo* (reperibile presso il Sito Web dell'Ateneo, nella sezione relativa alla documentazione: <http://di.unito.it/moduliateneo>), presentare un carico didattico secondo le norme descritte per il piano carriera. Ricordiamo che, nella definizione del carico didattico occorre rispettare le propedeuticità di frequenza (cioè non è possibile inserire esami di un dato anno se non si sono già inseriti in piano carriera tutti gli esami degli anni precedenti).

Nelle tabelle che compaiono in questa sezione sono riportati solo i dati essenziali dei corsi, che permettono di avere un quadro complessivo dei diversi anni. Maggiori dettagli (programma sintetico e docenti) si possono trovare nelle Appendici, oltre che, naturalmente, nel sito del Corso di Studi (<http://www.informatica.unito.it>), nella parte relativa alla "Offerta formativa".

La laurea triennale prevede 180 CFU. Nelle tabelle che seguono, che descrivono le unità didattiche, la somma totale dei CFU (primo anno in comune più secondo e terzo anno di ciascuno dei due curricula) è pari a 156. I restanti 24 CFU sono così suddivisi: 12 CFU per lo stage e la prova finale (si veda il paragrafo successivo) e 12 CFU liberi, a scelta dello studente.

Stage e prova finale

Per completare gli studi, lo studente dovrà effettuare un periodo di stage, la cui durata è di norma pari a tre mesi, presso un'azienda o un centro di ricerca, allo scopo di avere un contatto diretto con il mondo del lavoro (Le informazioni necessarie sono disponibili alla pagina <http://di.unito.it/stageccs>). Al termine dello stage, lo studente dovrà scrivere una relazione in cui si descriva l'attività svolta e i risultati ottenuti. Sia l'attività di stage che quella di scrittura della relazione verranno effettuate sotto la supervisione di un docente del Corso di Laurea e si concluderanno con un esame finale. Lo stage terminato con successo porta al conseguimento di 6 CFU mentre la Prova Finale, che include il lavoro di stesura della relazione del lavoro di stage e l'esame di laurea in senso stretto, è valutata in ulteriori 6 CFU.

CFU a scelta libera dello studente

Lo studente può scegliere i 12 CFU liberi all'interno dell'offerta formativa dell'intero Ateneo. La scelta deve essere congruente con l'obiettivo formativo della classe di laurea in Informatica e pertanto la scelta dei liberi deve essere convalidata da apposita commissione. Per ogni curricula troverete indicate le scelte per i CFU a scelta libera dello studente che sono automaticamente approvate, senza necessità di convalida.

Struttura dell'offerta formativa

Biennio propedeutico comune ai tre curricula

I anno

Codice	Insegnamento	CFU
mfn0570	Analisi Matematica	9
mfn0578	Matematica discreta e logica	12
mfn0582	Programmazione I	9
mfn0585	Programmazione II	9
mfn0586	Architettura degli elaboratori	9
mfn0588	Calcolo matriciale e ricerca operativa	6
mfn0590	Lingua Inglese I	3
	Totale 1 anno	57

II anno

Codice	Insegnamento	CFU
mfn0597	Algoritmi e strutture dati	9
mfn0598	Fisica	6
mfn0600	Elementi di probabilità e statistica	6
mfn0601	Sistemi operativi	12

mfn060 2	Basi di dati	9
mfn060 3	Linguaggi Formali e Traduttori	9
mfn060 4	Economia e gestione dell'impresa e diritto	9
	Totale 2 anno	60

Curriculum Informazione e Conoscenza

Il curriculum "*Informazione e conoscenza*" (IC) ha lo scopo di preparare laureati in grado di sviluppare e gestire sistemi informatici complessi, con particolare attenzione alle applicazioni Web. Il curriculum pone particolare attenzione allo sviluppo di applicazioni desktop e "mobile" di sistemi autonomi ed intelligenti, così come alla progettazione di architetture software robuste e flessibili, all'accesso efficiente ai dati ed alla loro analisi intelligente, volta a migliorare la fruizione delle grandi moli di informazioni disponibili online. Fornisce dunque una preparazione di base per rispondere alle esigenze della Società dell'Informazione, adatta alla continuazione degli studi nella Laurea Magistrale, affiancata ad una preparazione tecnica volta a preparare lo studente per un rapido inserimento nel mondo del lavoro.

III anno

Codice	Insegnamento	CFU
mfn060 5	Programmazione III	6
mfn060 6	Sviluppo delle applicazioni software	9
mfn060 7	Sistemi Intelligenti	6
mfn060	Interazione Uomo Macchina e	12

8	Tecnologie WEB	
mfn136 2	Reti I	6
	Crediti liberi a scelta dello studente	12
mfn064 5	Stage	6
mfn065 0	Prova finale	6
	Totale 3 anno	63

Possibili insegnamenti liberi che rendono il Piano di Studi automaticamente approvato secondo il curriculum Informazione e Conoscenza.

Codice	Insegnamento	CFU
mfn061 0	Linguaggi e Paradigmi di programmazione se non si è scelto l'insegnamento da 6 CFU	9
mfn135 4	Linguaggi e Paradigmi di programmazione se non si è scelto l'insegnamento da 9 CFU	6
mfn061 2	Calcolabilità e Complessità A	6
mfn061 7	Economia e gestione dell'innovazione	6
inf0072	Prolungamento Stage	6
mfn063 6	Sicurezza	6
mfn061 8	Sistemi Informativi	6
inf0003	Logica per l'Informatica	6
inf0004	Storia dell'Informatica	6

Curriculum Linguaggi e Sistemi

Il curriculum "*Linguaggi e Sistemi*" (LS) ha lo scopo di preparare laureati in grado di avere una solida preparazione nell'informatica, sia da un punto di vista fondazionale, sia per lo sviluppo di applicazioni.

Questo curriculum fornisce numerosi elementi formativi di base nel settore matematico e fisico e, per quanto riguarda il settore informatico, le competenze di base relative alle metodologie di programmazione e ai sistemi di elaborazione. Approfondisce inoltre i linguaggi di programmazione e i sistemi di calcolo.

III anno

Codice	Insegnamento	CFU
mfn0610	Linguaggi e paradigmi di programmazione	9
mfn0633	Metodi Formali dell'Informatica	9
mfn0606	Sviluppo delle applicazioni software	9
mfn1362	Reti I	6
mfn0605	Programmazione III	6
	Crediti liberi a scelta dello studente	12
mfn0645	Stage	6
mfn0650	Prova finale	6
	Totale 3 anno	63

Possibili insegnamenti liberi che rendono il piano di studi automaticamente approvato secondo il curriculum Linguaggi e Sistemi:

Codice	Insegnamento	CFU
mfn0617	Economia e gestione dell'innovazione	6
mfn0608	Interazione Uomo Macchina e tecnologie web se non si è scelto né mfn1353 né inf0002 né mfn0634	12
mfn1353	Interazione Uomo Macchina se non si è scelto mfn0608	6
inf0072	Prolungamento Stage	6

inf0002	Servizi Web se non si è scelto né mfn0608 né mfn0634	6
mfn063 6	Sicurezza	6
mfn061 8	Sistemi Informativi	6
mfn060 7	Sistemi intelligenti	6
mfn063 4	Tecnologie web se non si è scelto né inf0002 né mfn0608	6
inf0003	Logica per l'Informatica	6
inf0004	Storia dell'Informatica	6

Curriculum Reti e Sistemi Informatici

L'obiettivo di questo curriculum è quello di formare figure competenti nella progettazione, gestione e controllo di sistemi software. In quanto competenti in sistemi operativi, reti e sicurezza, linguaggi e ambienti di programmazione e basi dati, essi saranno in grado di gestire l'informatizzazione della piccola impresa e della pubblica amministrazione, e di inserirsi attivamente nei progetti delle grandi aziende per lo sviluppo e la gestione di sistemi software complessi.

III anno

Codice	Insegnamento	CFU
mfn0606	Sviluppo delle applicazioni Software	9
mfn0634	Tecnologie web	6
mfn0635	Reti di elaboratori	12
mfn0636	Sicurezza	6
	Insegnamenti a scelta	6
mfn0612	Calcolabilità e Complessità A (6 CFU)	6
mfn0605	Programmazione III - (6 CFU)	6

	Crediti liberi a scelta dello studente	12
mfn0645	Stage	6
mfn0650	Prova finale	6
	Totale 3 anno	63

Possibili corsi liberi che rendono il Piano di Studi automaticamente approvato secondo il curriculum Reti e Sistemi Informatici:

Codice	Insegnamento	CFU
mfn0610	Linguaggi e Paradigmi di programmazione se non si è scelto l'insegnamento da 6 CFU	9
mfn1354	Linguaggi e Paradigmi di programmazione se non si è scelto l'insegnamento da 9 CFU	6
mfn0617	Economia e gestione dell'innovazione	6
mfn1353	Interazione Uomo Macchina	6
inf0072	Prolungamento Stage	6
mfn0618	Sistemi Informativi	6
mfn0607	Sistemi Intelligenti	6
inf0003	Logica per l'Informatica	6
inf0004	Storia dell'Informatica	6

Struttura dell'offerta formativa rispetto al RAD della classe

Le tabelle che seguono descrivono l'impostazione dell'offerta formativa del nostro corso di studio (colonna "CFU") rispetto alla struttura della classe L-31 così come definita dal Ministero (Colonna "CFU RAD").

Attività di base

Ambito disciplinare	Settore	CFU	CFU RAD
Formazione matematico-fisica	MAT/01 Logica Matematica -- 6 CFU	27	18-30
	MAT/02 Algebra -- 6 CFU		
	MAT/05 Analisi Matematica -- 9 CFU		
	MAT/09 Ricerca Operativa -- 6 CFU		
Formazione informatica	INF/01 Informatica	27	24-36

Minimo di crediti riservati all'Ateneo (30 CFU da decreto ministeriale)

Totale attività di base	54	42-66
-------------------------	----	-------

Attività caratterizzante

Ambito: Discipline Informatiche		CFU	CFU Rad
Intervallo di crediti dell'ambito (min. 60CFU da decreto ministeriale)		78	69-90
Gruppo	Settore		
C11	INF/01 Informatica	78	69-90
C12		0	0-18

Minimo di crediti riservati all'Ateneo (60 CFU da decreto ministeriale)

Totale attività caratterizzanti	78	69-90
---------------------------------	----	-------

Attività affini e integrative

Ambito: attività formative affini o integrative		CFU	CFU Rad
Intervallo di crediti dell'ambito (<i>minimo 18CFU da decreto ministeriale</i>)		21	18-24
Gruppo	Settore		
A11	FIS/01 Fisica Sperimentale -- 6 CFU MAT/06 Probabilità e Statistica - 6 CFU	12-12	6-15
A12	IUS/02 Diritto Privato Comparato -- 3 CFU SECS-P/08 Economia e gestione delle imprese - 6 CFU	9-9	6-15

Totale attività affini	21	18-24
-------------------------------	----	-------

Altre attività

Ambito disciplinare		CFU	CFU Rad
A scelta dello studente		12	12-15
Per la prova finale e la lingua straniera (art. 10, comma 5, lettera d)	Per la prova finale	6	6-9
	Per la conoscenza della lingua straniera	3	2-6
Minimo numero di crediti riservati dall'Ateneo alle attività art. 10, comma 5, lett. c			-
Ulteriori attività formative (art.10, comma 5, lettera d)	Ulteriori conoscenze linguistiche	0	0-6
	Abilità informatiche e telematiche	0	0-6
	Tirocini formativi e di orientamento	6	0-6
	Altre conoscenze utili per l'inserimento nel mondo del lavoro	0	0-6
Minimo numero di crediti riservati dall'Ateneo alle attività art. 10, comma 5, lett. d			1
Per stages e tirocini presso imprese, enti pubblici o privati e ordini professionali		0	0-6

Totale Altre Attività	27	21-60
-----------------------	----	-------

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

Il patrimonio documentario è costituito da numerosi testi monografici e riviste specialistiche, in parte accessibili a scaffale aperto, in parte accessibili su richiesta del personale bibliotecario. La biblioteca dispone di una sala studio-consultazione dove è anche possibile effettuare ricerche bibliografiche online dai cataloghi accessibili da due postazioni informatiche.

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

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

Supporto on-line 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

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. Susanna Donatelli

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

Direttore del Dipartimento di Informatica: prof. Luca Console

e-mail: direttore@di.unito.it

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

Vicolo Benevello 3/a, Torino

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 delle due coorti precedenti che devono seguire i corsi del secondo e terzo anno rispettivamente. Sono inoltre riportati i corsi che verranno attivati nei prossimi anni, nonché l'elenco dei corsi esterni (vedi in seguito) e dei codici per le convalide dei corsi (per gli studenti in trasferimento).

Programmi, docenti e semestri per i corsi attivi nel 14/15.

La tabella che segue riporta tutti gli insegnamenti che saranno attivi nell'anno accademico 14/15. Nella tabella le colonne "Codice", "Insegnamento" e "SSD" indicano, rispettivamente, il codice di Ateneo, il titolo dell'insegnamento e il suo [settore scientifico disciplinare](#), "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. Tutti i corsi del primo e del secondo anno sono sdoppiati (corso A e corso B), mentre i laboratori del primo anno sono quadruplicati (turni A1, A2, B1 e B2) e quelli del secondo sono triplicati (T1, T2, T3). I programmi dei corsi sono invece presentati più avanti, a fine guida. Ricordiamo che analoghe informazioni sono reperibili sulle pagine web del corso di laurea (da <http://www.informatica.unito.it> seguire il link 'corsi', nella sezione "Informazioni per gli studenti").

Codice principale	INSEGNAMENT O	SSD	CFU	DOCENTI	Semestre	Note
mfn0597	Algoritmi e Strutture Dati	inf/01	9	Giovannetti (Teoria A) Esposito (lab T1), Damiani (Teoria B) Giovannetti (lab T2, lab T3)	2	
mfn0570	Analisi Matematica	mat/05	9	Viola (teoria A), docenti da definire (Eserc. A e B), Dambrosio (Teoria B), Costantini (pre-corso)	2	

mfn05 86	Architettura degli Elaboratori	inf/01	9	Gaeta (teoria A, lab A2), Boella (lab A1) Margaria (teoria B), Lucenteforte/Schifanella (lab B1), Donatelli/ Schifanella (lab B2)	2	
mfn06 02	Basi di Dati	inf/01	9	Demo (Teoria A, lab T1-A2), Anselma (lab T2), Pensa (Teoria B), Capecchi (lab T3)	2	
mfn061 2	Calcolabilità e Complessità A	inf/01	6	Ronchi della Rocca	2	Per il 14-15 mutuato su Metodi Formali dell'Informatica
mfn05 88	Calcolo Matriciale e Ricerca Operativa	mat/09	6	Grosso (A), Aringhieri (B)	1	
mfn06 04	Economia e Gestione dell'Impresa e Diritto	6 CFU secs- p/08 e 3 CFU ius/02	9	Pironti/Pisano (Economia A e B), Calliano (Diritto A e B)	2	
mfn061 7	Economia e Gestione dell'Innovazio ne	secs-p/08	6	Pironti/Pisano	1	
mfn06 00	Elementi di Probabilità e Statistica	mat/06	6	Garello (A e B condivisi), Esercitazioni separate per corso A e corso B, con docente da definire	1	
mfn059 8	Fisica	fis/01	6	Migliore (A) Pesando (B)	2	
mfn135 3	Interazione Uomo Macchina	inf/01	6	Sacco, Baldoni	1	mutua su modulo di Interazione Uomo Macchina e Tecnologie Web
mfn06 08	Interazione Uomo Macchina e Tecnologie Web	inf/01	12	Ardissono, Baldoni, Sacco	1	
mfn059 0	Lingua Inglese I	L-Lin/12	3	Griffin (esercitato-re), Radicioni, Cordero, Zacchi (responsabi-li)	1 e 2	
mfn061 0	Linguaggi e Paradigmi di	inf/01	9	Coppo, Dezani	1	

	Programmazione					
mfn1354	Linguaggi e Paradigmi di Programmazione	inf/01	6	Coppo, Dezani	2	mutua su modulo di Linguaggi e Paradigmi di Programmazione da 9 CFU
mfn0603	Linguaggi Formali e Traduttori	inf/01	9	Zacchi (Teoria A), Padovani (lab T1, lab T2), Dezani/Coppo (Teoria B), Sproston (lab T3)	1	
mfn0578	Matematica Discreta e logica	6 CFU mat/02 + 6 CFU mat/01	12	Docente da definire/Roggero (Mat Discr A), Rossi M. (Mat Discr B), Andretta (Log A), Cardone (Log B)	1	
mfn0633	Metodi Formali dell'Informatica	inf/01	9	Ronchi della Rocca	2	
mfn0582	Programmazione I	inf/01	9	Cardone (teoria A), De Pierro (lab A1-A2), Roversi (Teoria B, lab B1, lab B2)	1	
mfn0585	Programmazione II	inf/01	9	Bono (teoria A), Baroglio (lab A1-A2), Ardissono (teoria B), Bono (lab B1), Bettini (lab B2)	2	
mfn0605	Programmazione III	inf/01	6	Baldoni	1	
mfn0635	Reti di Elaboratori	inf/01	12	Sereno, Sirovich	1 e 2	
mfn1362	Reti I	inf/01	6	Botta	1	
inf0002	Servizi Web	inf/01	6	Ardissono	1	mutua su modulo di Interazione Uomo Macchina e Tecnologie Web
mfn0636	Sicurezza	inf/01	6	Bergadano	2	
mfn0618	Sistemi Informativi	inf/01	6	Meo	1	
mfn0607	Sistemi Intelligenti	inf/01	6	Torasso	2	
mfn0601	Sistemi Operativi	inf/01	12	Gunetti (Teoria A), Radicioni (lab T1 Unix), Micalizio (lab T2)	1	

				Unix), Gaeta (lab T2 C), Anselma (lab T2 C), Baroglio (Teoria B), Micalizio (lab T3 Unix e lab T3 C)	
mfn0606	Sviluppo delle Applicazioni Software	inf/01	9	Bono e Petrone (Teoria), Torta (lab 1), Picardi (lab 2)	2
mfn0634	Tecnologie Web	inf/01	6	Ruffo	1
mfn0645	Stage	altre attività	6		
mfn0650	Prova Finale	altre attività	6		

Programmi e docenti previsti per i corsi di futura attivazione:

Nell'A.A. 16/17, in coincidenza del terzo anno della coorte 2014, saranno attivati i seguenti corsi:

- Logica per l'Informatica, codice inf003, nel SSD mat/01
- Storia dell'Informatica, codice inf004, nel SSD inf/01

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

Logica per l'Informatica (prof. Andretta)

Il corso si propone di dare una solida preparazione di base in quelle aree della logica matematica che rivestono maggiore importanza per l'informatica. Nella prima parte si affronteranno i seguenti argomenti: Sintassi logica modale e logica temporale e loro applicazioni all'informatica; Linguaggi proposizionali e del prim'ordine; Deduzione naturale proposizionale e predicativa (in forma lineare alla Fitch e in forma arborea alla Gentzen); Calcolo dei seguenti e tableaux semantici; Sistemi formali. La seconda parte del corso verterà invece su: Semantica, Strutture del prim'ordine e la relazione di soddisfazione; Teorema di correttezza; Teorema di completezza (per linguaggi numerabili); Teorema di compattezza e modelli non standard dell'aritmetica. La terza parte del corso coprirà invece i seguenti argomenti: Logiche non classiche; Logica intuizionistica e modelli di Kripke; Cenni di logica modale e logica temporale e loro applicazioni all'informatica.

Storia dell'Informatica (proff. ri Cardone e Gunetti)

Il corso fornisce una introduzione alla storia dell'informatica, dedicando particolare attenzione ad alcune tappe che hanno fornito un contributo essenziale alla definizione della forma attuale dell'informatica (per esempio, lo sviluppo di Internet con il suo retroterra culturale e tecnologico).

L'interesse centrale del corso è per le idee e la loro evoluzione, più che per le singole innovazioni tecnologiche: non si tratta di una rassegna di modelli di macchina calcolatrice, ma di una introduzione ai modi di pensiero, alle metafore ed alle visioni che hanno caratterizzato la scienza dell'informazione e del calcolo attraverso la storia.

Il corso comprende, accanto ad un'introduzione istituzionale alla storia del calcolo basata su uno dei libri di testo elencati (il libro di Davis con alcuni complementi relativi ad argomenti non trattati in quel libro), anche una serie di lezioni monografiche dedicate a temi e lavori classici in varie aree dell'informatica.

Completeranno il corso una storia dell'evoluzione delle architetture, dei sistemi operativi e dei linguaggi di programmazione, insieme ad alcuni cenni di storia dell'informatica commerciale e di Internet.

L'esame del corso prevede, oltre ad un compito finale, l'esposizione di uno dei testi classici, a scelta dello studente, in forma di seminario.

Codici per corsi esterni.

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.

mfn105			
2	Attività Professionalizzanti	altre attività, a scelta lettera A	3

mfn150 2	Attività Professionalizzanti	altre attività, a scelta lettera A	3
-------------	------------------------------	------------------------------------	---

Codici per convalide di corsi e competenze.

Gli studenti che per effetto di passaggio o trasferimento abbia ottenuto la convalida di crediti liberi con nessuna corrispondenza specifica a corsi del CdS di Informatica, sono pregati di selezionare i codici contenitore MFN1522, MFN1409 e/o MFN1408 in base alla delibera della Commissione Passaggi e Trasferimenti, integrando eventualmente con altro corso libero da selezionare dalla lista prevista per il curriculum scelto, in caso serva ancora raggiungere il minimo di 12 CFU..

mfn140 9	Altre Attività	altre attività, a scelta lettera A	12
mfn140 8	Altre Attività	altre attività, a scelta lettera A	6
mfn152 2	Altre Attività	altre attività, a scelta lettera A	6

Informazioni aggiornate al 12 ottobre 2016

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

Syllabus

N° documenti: 26

Testi del Syllabus

Docente	GIOVANNETTI ELIO	Matricola: 013049
Anno offerta:	2014/2015	
Insegnamento:	MFN0597 - ALGORITMI E STRUTTURE DATI (Algorithms and Data Structures)	
Corso di studio:	008707 - 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

Lingua insegnamento

Italiano.

Contenuti

Problemi e algoritmi: risolubilità, correttezza, complessità.

Primi esempi di problemi e algoritmi.

Complessità degli algoritmi ricorsivi e equazioni di ricorrenza.

Eliminazione della ricorsione di coda: tecniche ed esempi di eliminazione (algoritmo di Euclide, esponenziale veloce, torri di Hanoi, visita semi-iterativa di alberi, ecc.)

Strutture-dati e tipi astratti.

Code con priorità e heap: il tipo astratto coda con priorità, realizzazioni ingenuie, realizzazione tramite la struttura heap.

Il problema dell'ordinamento e gli algoritmi di ordinamento e selezione.

Algoritmi di ordinamento lineari (non per confronti).

Dizionari, alberi di ricerca, tabelle hash.

Cenni alle realizzazioni di tipi astratti e strutture-dati (Pila, Coda, Lista, Insieme, Bag, Mappa, Tabella hash, ecc.) nella libreria standard Java.

Strutture union-find.

Il problema dello string-matching: algoritmo ingenuo; algoritmo di Knuth-Morris-Pratt.

Algoritmi greedy.

Tecnica algoritmica di ricorsione con memoizzazione e cenni di programmazione dinamica.

Grafi

Rappresentazione dei grafi.

Visite in ampiezza e profondità.

Componenti fortemente connesse.

Test di aciclicità.

Ordinamento topologico.

Cammini minimi: algoritmo di Dijkstra.

Minimo albero di copertura: algoritmo di Prim, algoritmo di Kruskal.

Altri algoritmi sui grafi.

Correttezza e complessità degli algoritmi sui grafi.

Testi di riferimento

Testi principali:

Crescenzi, Gambosi, Grossi: Strutture di dati e algoritmi. Pearson Addison-Wesley, Seconda edizione, 2012.

Horstmann: Concetti di informatica e fondamenti di Java, Quinta Edizione (o successiva), Apogeo, 2010, euro 45.

Testi per eventuali approfondimenti:

Demetrescu, Finocchi, Italiano: Algoritmi e strutture dati, Seconda edizione, McGraw-Hill, 2008.

Bertossi, Montresor, Algoritmi e Strutture di Dati, Città Studi Edizioni, Torino, 2010.

Obiettivi formativi

Introdurre alcuni degli algoritmi e delle strutture-dati fondamentali e di fornire gli strumenti metodologici di base per la risoluzione di problemi algoritmici mediante: (i) il

Tipo testo

Testo

progetto, l'analisi, e il confronto di algoritmi e strutture dati; e (ii) la loro implementazione in un linguaggio orientato agli oggetti (Java), ed eventualmente in un linguaggio imperativo non orientato agli oggetti (C).

Prerequisiti

Tutte le nozioni e competenze fornite dai corsi del primo anno e dal corso di Sistemi Operativi (in particolare la conoscenza del linguaggio C).

Propedeutico a

I corsi del terzo anno.

Note

Supporto on-line tramite la piattaforma MOODLE.

Modalità di verifica dell'apprendimento

L'esame è costituito di due parti:

1. Una prova scritta, contenente domande ed esercizi sugli argomenti presentati nel corso (inclusi gli esercizi assegnati in laboratorio).
2. Un colloquio consistente in:
 - una discussione della prova scritta;
 - una discussione e valutazione, alla macchina, degli esercizi laboratorio svolti durante il corso;
 - domande sugli argomenti svolti nel corso.

Risultati di apprendimento attesi

Capacità di realizzare componenti software che supportino le strutture dati fondamentali e i relativi algoritmi usando i linguaggi Java e C, sfruttando in modo opportuno le librerie standard di questi linguaggi.

Frequenza

Fortemente raccomandata



Testi in inglese

Tipo testo

Testo

Lingua insegnamento

Italian.

Contenuti

Problems and algorithms: solvability, correctness, complexity.

First examples of problems and algorithms.

Analysis of recursive algorithms through recurrence equations.

Tail-recursion elimination: techniques and examples (Euclid's algorithm, fast exponential, Tower of Hanoi, semi-iterative tree traversal, etc.)

Data structures and Abstract Data Types.

Priority queues and heaps: the ADT priority queue, naive implementations, implementation with the heap structure.

Sorting and selection.

Linear sorting algorithms.

Dictionaries, search trees, hash tables.

ADTs and data structures in Java API Library (Stack, Queue, List, Set, Bag, Map, Hashtable, etc.).

Union-find.

String-matching: the naive algorithm; Knuth-Morris-Pratt algorithm.

Greedy algorithms.

Recursion with memoisation and elementary dynamic programming.

Graphs

Graph representation: adjacency lists, adjacency matrix.

Breadth-first traversal and depth-first traversal.

Shortest path and Dijkstra's algorithm. All-pairs shortest paths.

Minimum spanning tree: Prim's algorithm, Kruskal's algorithm.

Edge classification induced by depth-first traversal.

Aciclicity test.

Topological sorting.

Finding the strongly connected components.

Other graph algorithms.

Correctness and complexity of graph algorithms.

Testi di riferimento

Main textbooks:

Crescenzi, Gambosi, Grossi: *Strutture di dati e algoritmi*. Pearson Addison-Wesley, Seconda edizione, 2012.

Horstmann: *Concetti di informatica e fondamenti di Java*, Quinta Edizione (o successiva), Apogeo, 2010, euro 45.

Other textbooks that might be useful:

Demetrescu, Finocchi, *Italiano: Algoritmi e strutture dati*, Seconda edizione, McGraw-Hill, 2008.

Bertossi, Montresor, *Algoritmi e Strutture di Dati*, Città Studi Edizioni, Torino, 2010.

Obiettivi formativi

To introduce some of the fundamental algorithms and data structures and to provide the basic methodological tools for solving algorithmic problems by means of: (i) project, analysis, and comparison of algorithm and data structures, and

Tipo testo**Testo**

(ii) their implementation in an object oriented language (Java) and possibly also in a not object-oriented imperative language (C).

Prerequisiti

All the notions and competencies given by the courses of the first year and by the course of Sistemi Operativi (in particular the knowledge of the C language).

Propedeutico a

The courses of the third year.

Note

On-line support via the MOODLE platform.

Modalità di verifica dell'apprendimento

The exam consists of two parts:

1. a written part, consisting of questions and exercises on the material presented in the course (including the exercises assigned in laboratory).

2. A oral discussion consisting of:

- discussion of part 1;
- discussion and evaluation, on the computer, of the exercises assigned in laboratory during the course;
- questions on the material presented in the course.

Risultati di apprendimento attesi

Ability to implement software components supporting fundamental data structures and the associated algorithms by using the languages Java and C, by suitably exploiting the standard libraries of these languages.

Frequenza

Strongly recommended

Testi del Syllabus

Docente	VIOLA GABRIELLA	Matricola: 013574
Anno offerta:	2014/2015	
Insegnamento:	MFN0570 - ANALISI MATEMATICA (Mathematical Analysis)	
Corso di studio:	008707 - INFORMATICA (Computer Science)	
Anno regolamento:	2014	
CFU:	9	
Settore:	MAT/05	
Tipo attività:	A - Base	
Partizione studenti:	-	
Anno corso:	1	
Periodo:	Secondo Semestre	

Testi in italiano

Tipo testo	Testo
Lingua insegnamento	Italiano
Contenuti	Numeri reali, retta orientata, piano cartesiano. Funzioni elementari: potenze, esponenziali, logaritmiche, trigonometriche. Successioni e serie numeriche. Limiti e continuità per funzioni reali di variabile reale. Calcolo differenziale e applicazioni. Sviluppi di Taylor e confronto tra infinitesimi. Il problema dell'area: l'integrale definito. il Teorema Fondamentale del Calcolo integrale. Metodi numerici per il calcolo di integrali definiti. Integrali impropri su intervalli illimitati.
Testi di riferimento	M. Bramanti, C.D. Pagani, S. Salsa Analisi Matematica 1 - Zanichelli Editore P. Marcellini, C. Sbordone Esercitazioni di matematica (Vol. 1 - Parti 1-2) - Liguori Editore
Obiettivi formativi	Lo studente deve saper: effettuare semplici calcoli di limiti, derivate, integrali; disegnare il grafico qualitativo di una funzione reale di variabile reale; studiare la convergenza di serie numeriche e integrali impropri; effettuare ragionamenti elementari di tipo logico-deduttivo sui risultati classici dell'analisi matematica di base.
Prerequisiti	Conoscenze matematiche di base fornite dalla scuola superiore. In particolare la conoscenza dei seguenti argomenti: sistemi di equazioni e disequazioni algebriche, razionali fratte, irrazionali, esponenziali, logaritmiche e trigonometriche. Il precorso (prima settimana di lezione) fornisce le "competenze attese in ingresso".
Propedeutico a	Calcolo delle Probabilità e Statistica (2° anno); Complementi di Analisi e Probabilità (4°, 5° anno);
Note	Durante il corso si controlla l'apprendimento degli studenti mediante le domande poste durante le ore di lezione e i ricevimenti, e attraverso la correzione degli esercizi assegnati. Vengono inoltre forniti esercizi, testi degli esami scritti precedenti, eventuali note aggiuntive su quanto presentato a lezione. Queste risorse sono disponibili online.
Modalità di verifica dell'apprendimento	<p>L'esame consiste in una prova scritta ed una orale (entrambe obbligatorie per tutti gli studenti, indipendentemente dall'Anno Accademico di iscrizione). Per accedere alla prova orale è necessario aver ottenuto un voto di almeno 18/30 nella prova scritta. Lo studente che abbia superato una prova scritta deve obbligatoriamente presentarsi alla prima prova orale utile; in caso di prova orale insufficiente deve sostenere di nuovo anche la prova scritta. La validità di una prova scritta superata è quindi limitata alla sola prova orale immediatamente successiva.</p> <p>La prova scritta ha una durata di due ore e trenta minuti; durante la prova non è consentito l'uso di calcolatrici e altri strumenti elettronici. Non è permesso consultare testi o appunti; lo studente può portare con sé un foglio in formato A4 con le formule che ritenga utili.</p> <p>La prova orale verte sugli argomenti trattati a lezione ed esercitazioni.</p> <p>Gli studenti degli anni accademici passati sostengono la stessa prova scritta degli studenti in corso; alla prova orale possono scegliere se essere interrogati sul programma dell'anno in corso o sul programma dell'anno in cui si sono immatricolati (in questo caso devono presentarsi con copia del programma). La prova orale è obbligatoria per tutti gli studenti, indipendentemente dalle modalità d'esame dell'anno in cui si sono immatricolati.</p>

Tipo testo

Testo

Risultati di apprendimento attesi

Per quanto riguarda il sapere, si veda il programma del corso. Per quanto riguarda le capacità, si vuole che lo studente sappia svolgere correttamente alcuni semplici esercizi che coinvolgono i concetti di base (calcolo di limiti, derivate e integrali, studio di funzioni, studio di equazioni numeriche non risolubili direttamente, studio di serie). Ci si aspetta inoltre che il corso di Analisi, assieme agli altri corsi di matematica del primo anno, contribuisca a far acquisire allo studente alcune capacità logiche di base (distinguere tra le ipotesi e la tesi di un teorema, ad esempio).

Frequenza

La frequenza alle lezioni non è obbligatoria ma fortemente consigliata per una soddisfacente comprensione della materia



Testi in inglese

Tipo testo	Testo
Lingua insegnamento	Italian
Contenuti	Real numbers, oriented line, cartesian plane. Elementary functions: powers, roots, exponentials, logarithmics, trigonometrics. Sequences and series. Limits and continuity for functions of one variable. Differential calculus and applications, Taylor expansions. Integral calculus; the definite integral and its geometrical meaning. The fundamental theorem of integral calculus. Approximate numerical calculation of an integral. Generalized integrals over non limited intervals.
Testi di riferimento	M. Bramanti, C.D. Pagani, S. Salsa <i>Analisi Matematica 1</i> - Zanichelli Editore P. Marcellini, C. Sbordone <i>Esercitazioni di matematica (Vol. 1 - Parti 1-2)</i> - Liguori Editore
Obiettivi formativi	The student must be able to: perform simple calculations of limits, derivatives, integrals; draw the graph of a real function; study the convergence of numerical series and improper integrals; elementary reasoning to make logical-deductive on the classical results of basic calculus
Prerequisiti	Basic mathematical knowledge provided by the secondary school. Particularly is required confidence with systems of algebraic, rational, irrational, exponentials, logarithmic and trigonometric equations and inequalities,. In the first week of the course will be provided all the expected competences.
Propedeutico a	Probability and Statistics (2nd year); Complementary Calculus and Probability (4th, 5th year);
Note	Students' progress will be checked during the course through questions asked in class and office hours, and across the correction of assignments. Exercises, written examinations of earlier sessions and additional notes will be available online.
Modalità di verifica dell'apprendimento	The examination consists of a written and an oral tests (both compulsory for all students, regardless of the Academic enrollment). To access the oral test, one must have obtained a mark of at least 18/30 in the written test. The validity of a written test is limited to the immediately following oral test. Therefore, students who have passed a written test must necessarily attend the subsequent oral test; in case of inadequate oral test, one must retake the written test. The written test has a duration of 2h30'; during the test the use of calculators and other electronic devices is forbidden. It's not allowed to refer to notes or texts but the students can carry a A4 sheet with relevant formulas. The oral test focuses on topics covered in class and tutorials. The students of past academic years are required to attend the current written test; for the oral test they may choose to be interviewed on the program for the current year or the one in which they are registered (in this case they must present a copy of the program). The oral test is compulsory for all students, regardless of the year in which they are registered.
Risultati di apprendimento attesi	With regard to knowledge, see the course syllabus. Regarding expected acquired skills, the student must be able to properly carry out tasks that involve the basic concepts (Calculation of limits, derivatives and integrals, functions' graphs, numerical equations can not

Tipo testo**Testo**

be solved directly, numerical series).

It is also expected that this course, along with the other mathematical ones of the first year, will help the student to handle some basic logic skills (distinguish between the hypothesis and the thesis of a theorem, for instance).

Frequenza

Attendance is not mandatory but strongly recommended for a satisfactory understanding of the subject

Testi del Syllabus

Docente	MARGARIA INES MARIA	Matricola: 010621
Anno offerta:	2014/2015	
Insegnamento:	MFN0586 - ARCHITETTURA DEGLI ELABORATORI (Computer architecture)	
Corso di studio:	008707 - INFORMATICA (Computer Science)	
Anno regolamento:	2014	
CFU:	9	
Settore:	INF/01	
Tipo attività:	A - Base	
Partizione studenti:	-	
Anno corso:	1	
Periodo:	Secondo Semestre	

Tipo testo	Testo
Lingua insegnamento	Italiano
Contenuti	<ol style="list-style-type: none">1. Introduzione all'organizzazione strutturata dei calcolatori: macchine virtuali, evoluzione storica dei calcolatori.2. Struttura dei calcolatori, macchina di von Neumann.3. Codifica binaria dell'informazione: numeri assoluti e frazionari, numeri relativi, numeri a virgola mobile, codifica di caratteri, immagini, suoni.4. Livello logico digitale: porte logiche e algebra di Boole, circuiti logici di base, latch, flip-flop, registri, chip di memoria.5. Memoria: organizzazione della memoria principale, memoria cache, memorie secondarie, bus.6. Il livello della microarchitettura: organizzazione della CPU ed esecuzione delle istruzioni, l'architettura di Mic-1, il livello ISA della IJVM, un microprogramma per Mic-1, miglioramento delle prestazioni: Mic-2 e Mic-3.7. Il livello ISA: caratteristiche delle istruzioni macchina, modelli di memoria, modalità di indirizzamento, formati delle istruzioni, architetture RISC e CISC, istruzioni di I/O, interrupt e trap.8. Assembler, linker, loader. <p>Il corso prevede, oltre a lezioni teoriche, una serie di lezioni e esercitazioni in laboratorio sui circuiti logici, e sulla microprogrammazione e sui linguaggi assemblativi, con l'ausilio di appositi applicativi.</p>
Testi di riferimento	<p>Andrew S. Tanenbaum, Todd Austin. Architettura dei calcolatori 6/Ed. Un approccio strutturale. Pearson Education Italia, 2013. ISBN 9788871929620</p> <p>oppure</p> <p>Andrew S. Tanenbaum. Architettura dei calcolatori: un approccio strutturale. 5a Edizione. Pearson - Addison Wesley, 2006. ISBN 8871922719.</p>
Obiettivi formativi	<p>Fornire allo studente:</p> <ul style="list-style-type: none">- la conoscenza dell'organizzazione hardware degli elaboratori, attraverso la nozione di gerarchie di macchine virtuali;- la comprensione delle funzioni svolte dall'hardware e utilizzate dai sistemi operativi;- le nozioni base per la comprensione del processo di traduzione da linguaggio ad alto livello a linguaggio macchina.
Propedeutico a	Il corso è fortemente consigliato per la comprensione dei corsi di Linguaggi Formali e Traduttori e Sistemi Operativi.
Modalità di verifica dell'apprendimento	Esame scritto.



Testi in inglese

Tipo testo

Testo

Lingua insegnamento

Italian

Contenuti

1. Introduction to structured computer organization: computers as virtual machines, evolution of multilevel machines.
2. Computer System Organization and von Neumann machine.
3. Binary coding of information: natural numbers, fractions, negative numbers, floating-point numbers, character codes, images, sounds.
4. Logic level: logic gates and Boolean algebra, basic digital logic circuits, latches, flip-flops, registers, memory chips.
5. Memory: RAM organization, cache memory, secondary storage, bus.
6. Microarchitecture level: CPU organization, the architecture of Mic-1, the ISA level of IJVM, a microprogram for Mic-1, performance improvement: Mic-2 and Mic-3.
7. The ISA level: machine instructions, memory models, addressing, instruction formats, RISC and CISC architectures, instructions for I/O, interrupts and traps.
8. Assembler, linker, loader.

In addition to lectures, the course includes the implementation of exercises on logic circuits, and microprogramming and the assembly language, by means of suitable software.

Testi di riferimento

Andrew S. Tanenbaum, Todd Austin. Architettura dei calcolatori 6/Ed. Un approccio strutturale. Pearson Education Italia, 2013. ISBN 9788871929620

or

Andrew S. Tanenbaum. Architettura dei calcolatori: un approccio strutturale. 5a Edizione. Pearson - Addison Wesley, 2006. ISBN 8871922719.

Obiettivi formativi

Provide the student with:

- knowledge of computer hardware organization, through the notion of hierarchy of virtual machines;
- comprehension of the hardware functions, used by the operating systems;
- basic notions for understanding the process of translation from high-level language to machine language.

Propedeutico a

The course is highly recommended for the comprehension of the courses: Formal Languages and Translators, Operating Systems.

Modalità di verifica dell'apprendimento

Written.

Testi del Syllabus

Docente	DEMO GIUSEPPINA	Matricola: 011499
Anno offerta:	2014/2015	
Insegnamento:	MFN0602 - BASI DI DATI (Databases)	
Corso di studio:	008707 - 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
Lingua insegnamento	Italiano
Contenuti	<p>Funzionalità dei sistemi di gestione di basi di dati. Fondamenti teorici delle basi di dati relazionali:</p> <ul style="list-style-type: none">il modello relazionale delle basi di dati (definizioni, proprietà principali, vincoli di integrità),algebra relazionale,introduzione al calcolo relazionale,introduzione a SQL (DDL e DML),dipendenze funzionali e teoria della normalizzazione,memorizzazione efficiente dei dati (B+ alberi),cenni alle tecniche di ottimizzazione,Introduzione alle transazioni: problemi di concorrenza e di affidabilità, livelli di isolamento. <p>Progettazione e programmazione delle basi dati:</p> <ul style="list-style-type: none">Specifiche d'interrogazioni ed realizzazione in SQL (con esercitazioni in laboratorio su ORACLE).Catalogo di un DBMS (con esercitazioni in laboratorio su ORACLE).Definizione e ruolo di "database administrator".Cenni ai meccanismi d'autorizzazione offerti da SQL (con esercitazioni in laboratorio su ORACLE).Introduzione alle metodologie di progettazione del software e loro relazione con la progettazione della basi di dati, argomenti non trattati e relazione con altri corsi (es. Ingegneria del Software).Progettazione concettuale/logica, usando il modello ER (Entity Relationship).Considerazioni sui parametri quantitativi dello schema logico.Considerazioni su meccanismi d'indicizzazione.
Testi di riferimento	<p>Atzeni, Ceri, Fraternali, Paraboschi, Torlone, "Basi di dati - Modelli e linguaggi di interrogazione", McGraw-Hill, Quarta edizione, 2013.</p> <p>Materiali aggiuntivi sono forniti dai docenti.</p>
Obiettivi formativi	<p>Il corso è un'introduzione alle basi di dati e ai sistemi di gestione delle medesime. Si propone perciò di fornire allo studente le prime competenze teoriche e pratiche sul modello relazionale, sulla progettazione di una base di dati e sull'architettura dei sistemi di gestione. In aula saranno introdotti i concetti teorici di base. Gli studenti acquisiranno conoscenze (teoriche e pratiche) su: progetto di una base dati, modello Entità Associazioni (EA) (o Entity Relationship), modello relazionale, algebra relazionale e SQL, dipendenze funzionali e normalizzazione, elementi di architettura dei gestori di basi di dati relazionali (concorrenza, ripristino e buffer management, dizionario dei dati, memorizzazione efficiente dei dati), trigger, stored procedure. In laboratorio gli studenti lavoreranno al progetto di una base dati e ad esercitazioni su casi di studio in SQL.</p>
Prerequisiti	Corsi di Logica, Matematica Discreta e Programmazione 1 e 2
Propedeutico a	Corsi di Sistemi informativi, Servizi web e Tecnologie web

Tipo testo

Modalità di verifica dell'apprendimento

Testo

E' OBBLIGATORIA l'iscrizione all'ambiente di e-learning MOODLE dove gli studenti troveranno forum di discussione, materiale didattico e compiti assegnati e quiz sugli argomenti delle lezioni preparati dai docenti.

La modalita' d'esame e' scritta:

Prova scritta: con domande sulla teoria ed esercizi simili a quelli svolti in aula ed in laboratorio, al fine di verificare l'apprendimento dei concetti di base.

Il voto finale viene calcolato integrando i voti di scritto e laboratorio. La prova scritta e la discussion del laboratorio non hanno regole di precedenza se non che devono essere sostenute nella stessa sessione di esami.

Risultati di apprendimento attesi

Gli studenti devono aver assimilato i concetti:

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

Inoltre lo studente deve:

saper analizzare i requisiti per la progettazione di una base di dati per rappresentarli in forma di diagrammi Entity Relationship o Entita' Associazioni (EA)essere in grado di trasformare schemi EA in schemi relazionali,

essere in grado di trasformare interrogazioni in forma testuale in interrogazioni SQL, essere in grado di eseguire semplici ristrutturazioni di schemi EA sulla base di dati quantitativi (ad esempio: numero stimato di record e criticita' delle interrogazioni,

aver compreso i concetti e l'uso di transazioni, trigger e stored procedure.

Frequenza

La frequenza costante è caldamente consigliata.



Testi in inglese

Tipo testo

Testo

Contenuti

Introduction to databases, database management systems and database design.

Main titles:

- relational model and its main properties
- relational algebra and calculus
- SQL (in deep)
- functional dependencies and normal forms (BCNF, 3FN)
- physical structures, B-tree
- introduction to logical optimisation, concurrency and recovery.

As for database design:

- conceptual design
- logical and physical design
- reverse modelling.

Testi di riferimento

Atzeni, Ceri, Fraternali, Paraboschi, Torlone, "Basi di dati - Modelli e linguaggi di interrogazione", McGraw-Hill, Fourth edition, 2013.

Materiali aggiuntivi sono forniti dai docenti.

Obiettivi formativi

Students are introduced to databases (db) and to db systems. They will acquire basic theoretical and practical competencies on the relational model and languages, on db design activity and on a db system architecture.

-

Prerequisiti

Courses of;
Logic,
Discrete mathematics and Programming 1 and 2.

Propedeutico a

Courses of;
Information systems,
Web services, Web Technologies.

Modalità di verifica dell'apprendimento

The students must enroll to the moodle environment

Frequenza

Attendance is not mandatory but warmly recommended

Testi del Syllabus

Docente	GROSSO ANDREA CESARE	Matricola: 014296
Anno offerta:	2014/2015	
Insegnamento:	MFN0588 - CALCOLO MATRICIALE E RICERCA OPERATIVA (Matrix Calculus and Operational Research)	
Corso di studio:	008707 - INFORMATICA (Computer Science)	
Anno regolamento:	2014	
CFU:	6	
Settore:	MAT/09	
Tipo attività:	A - Base	
Partizione studenti:	-	
Anno corso:	1	
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 a cura del 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.).
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.
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	PIRONTI MARCO	Matricola: 014125
Anno offerta:	2014/2015	
Insegnamento:	MFN0604A - ECONOMIA E GESTIONE DELL'IMPRESA E DIRITTO / A (Management, business administration and computer law A)	
Corso di studio:	008707 - INFORMATICA (Computer Science)	
Anno regolamento:	2013	
CFU:	6	
Settore:	SECS-P/08	
Tipo attività:	C - Affine/Integrativa	
Partizione studenti:	-	
Anno corso:	2	
Periodo:	Secondo Semestre	

Testi in italiano

Tipo testo	Testo
Lingua insegnamento	Italiano
Contenuti	<p>Il corso è incentrato sui concetti basi del management aziendale, utili per far prendere al partecipante dimestichezza con il mondo del management e della strategia aziendale. Il corso svilupperà anche il modello canvas utile per lo sviluppo di nuovi progetti aziendali e per la coordinazione di tutte le funzioni che ruotano intorno ad un progetto.</p> <p>l'analisi del canvas la proposizione di valore la definizione del consumatore la relazione con il consumatore i canali distributivi i flussi di ricavo i flussi di costi le RISORSE chiave e attività chiave i PARTNERS l'analisi finanziaria.</p>
Testi di riferimento	<p>The Startup Owners Manual: by Steve blank The Lean Startup: by Eric Ries Il Business Plan: by Antonio Borello</p>
Obiettivi formativi	<p>L'obiettivo del corso è formalizzare il percorso che porta dalla nascita dell'idea alla creazione di un'azienda. Durante il corso i partecipanti imparano quali sono i passi fondamentali per la creazione di una start up, e quali le differenze tra una start up e una azienda già creata. Obiettivo di questo corso è anche quello di ottenere gli strumenti per la valutazione del progetto</p>
Propedeutico a	Nulla
Modalità di verifica dell'apprendimento	<p>La verifica dell'apprendimento avverrà attraverso un esame scritto in aula e una tesina sviluppata a gruppi all'interno del corso.</p>



Testi in inglese

Tipo testo	Testo
Lingua insegnamento	Italian
Contenuti	<p>The course focuses on the basics of business management concepts, useful to make the participant familiar with the world of management and business strategy. The course will also develop the canvas model used to new business projects development and to coordinate all functions that revolve around a project.</p> <p>SYLLABUS</p> <ul style="list-style-type: none">the analysis of the canvasthe value propositionthe definition of the consumerthe relationship with the consumerdistribution channelsrevenue STREAMCOST STREAMKEY RESOURCE AND KEY ACTIVITIESpartnersfinancial analysis
Testi di riferimento	<p>The Startup Owners Manual: by Steve blank The Lean Startup: by Eric Ries Il Business Plan: by Antonio Borello</p>
Obiettivi formativi	<p>The objective of the course is to formalize the path that leads from the birth of the idea to the creation of a company. During the course, participants learn what are the basic steps for creating a start-up, and what are the differences between a start-up company and one already created. The objective of this course is also to get the tools for the evaluation of the project</p>
Propedeutico a	Nothing
Modalità di verifica dell'apprendimento	A written assessment will be required to test the learning level

Testi del Syllabus

Docente	PIRONTI MARCO	Matricola: 014125
Anno offerta:	2014/2015	
Insegnamento:	MFN0617 - ECONOMIA E GESTIONE DELL'INNOVAZIONE (Managing Innovation)	
Corso di studio:	008707 - INFORMATICA (Computer Science)	
Anno regolamento:	2012	
CFU:	6	
Settore:	SECS-P/08	
Tipo attività:	D - A scelta dello studente	
Partizione studenti:	-	
Periodo:	Primo Semestre	

Tipo testo	Testo
Lingua insegnamento	Italiano
Contenuti	<p>La crescita di un'azienda o di un paese attraverso lo sviluppo di innovazione sembra essere davvero difficile. Benché molte aziende si sforzano di ottenere rendimenti positivi dal loro investimento in innovazione mancano gli strumenti per rendere l'innovazione parte della pratica quotidiana. Questo corso vi aiuterà a vedere e superare le trappole che si insinuano nel modo di innovazioni di successo, presentando strumenti, framework e modelli come:</p> <ul style="list-style-type: none">• Individuare le opportunità che gli altri potrebbero perdere• Creare i sistemi e le strutture per strutturare business vincenti• Strutturare soluzioni per nuovi business che possano contrastare i leader di mercato. <p>Questo corso ti insegnerà come l'innovazione possa essere creata, gestita e guidata se si conoscono gli strumenti:</p> <ul style="list-style-type: none">• identificare le opportunità• definire le ideeo sviluppare idee innovativeo definire una strategia coerente• strutturare le attività economicheo guidare il progetto emergente attraverso la strategia correttao creare e gestire un team innovativo• costruire la capabilityo organizzare l'azienda per innovareo metriche di innovazione
Testi di riferimento	<p>Paola Maria Pisano (2010). "Managing innovation: creare, gestire e diffondere innovazione nei sistemi relazionali" Padova, libreriauniversitaria.it, ISBN: 9788862920575</p> <p>Henry, Jane & Mayle, David (2002 second edition) Managing Innovation and Change</p> <p>Sarah A Meyer (2000) "Managing within a creative environment" Design Management Journal Vol 11 no 3 Summer</p> <p>Quinn, J. B. 1985. Managing innovation: Controlled chaos. Harvard Business Review, 63 (May-June): 78-84.</p> <p>Tidd, Joe, Beasant, John and Pavitt, Keith. 2001 2nd Edition. Managing Innovation - Integrating Technological, Market and Organizational Change. Wiley.</p>
Obiettivi formativi	<ol style="list-style-type: none">1) Analizzare le metodologie per la creazione di innovazione2) Analizzare il processo di sviluppo dell'innovazione3) Analizzare e sviluppare casi pratici per capire l'applicazione della teoria
Prerequisiti	EGI
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.

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

Non obbligatorio

Testi in inglese

Tipo testo	Testo
Lingua insegnamento	Italian
Contenuti	<p>Achieving growth through innovation seems so difficult. Almost every company struggles to achieve positive returns on its innovation investment . Some simply lack the tools to make innovation part of everyday practice. This course will help you see and overcome the traps that stand in the way of successful innovation, presenting frameworks tools and templates that will help innovators like you to:</p> <ul style="list-style-type: none">Spot opportunities that others are likely to missCraft solutions that create new growth while disrupting the current market leaderManage the risk of early stage projectsBuild systems and structures to create winning businesses again and again . This course will help you see that innovation can be managed and mastered. <p>syllabus:</p> <ul style="list-style-type: none">• identify opportunities• formulate and shape ideaso developing disruptive ideaso assessing a Strategy's Fit with a Pattern• build the businesso Mastering Emergent Strategieso Assembling and Managing Project Teams• Build Capabilitieso Organizing to innovateo Innovation Metrics
Testi di riferimento	<p>Paola Maria Pisano (2010). "Managing innovation: creare, gestire e diffondere innovazione nei sistemi relazionali" padova, libreriauniversitaria.it, ISBN: 9788862920575</p> <p>Henry, Jane & Mayle, David (2002 second edition) Managing Innovation and Change</p> <p>Sarah A Meyer (2000) "Managing within a creative environment" Design Management Journal Vol 11 no 3 Summer</p> <p>Quinn, J. B. 1985. Managing innovation: Controlled chaos. Harvard Business Review, 63 (May-June): 78-84.</p> <p>Tidd, Joe, Beasant, John and Pavitt, Keith. 2001 2nd Edition. Managing Innovation - Integrating Technological, Market and Organizational Change. Wiley.</p>
Obiettivi formativi	<ol style="list-style-type: none">1) Define the methodology for innovation creation and developing2) Using practice case to apply the theory.
Prerequisiti	EGI
Propedeutico a	Nothing
Modalità di verifica dell'apprendimento	A written assessment will be required to test the learning level

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

Not mandatory

Testi del Syllabus

Docente	GARELLO GIANLUCA	Matricola: 013419
Anno offerta:	2014/2015	
Insegnamento:	MFN0600 - ELEMENTI DI PROBABILITA' E STATISTICA (Foundations of Probability and Statistics)	
Corso di studio:	008707 - INFORMATICA (Computer Science)	
Anno regolamento:	2013	
CFU:	6	
Settore:	MAT/06	
Tipo attività:	C - Affine/Integrativa	
Partizione studenti:	-	
Anno corso:	2	
Periodo:	Primo Semestre	



Testi in italiano

Tipo testo

Testo

Lingua insegnamento

Italiano

Contenuti

INTRODUZIONE ALLA STATISTICA

Raccolta dei dati e statistica descrittiva. Popolazioni e campioni,

STATISTICA DESCRITTIVA

Organizzazione e descrizione dei dati: tabelle e grafici delle *frequenze assolute e delle frequenze relative; raggruppamento dei dati, istogrammi, ogive e diagrammi stem and leaf. Le grandezze che sintetizzano i dati: media, mediana e moda campionarie; varianza e deviazione standard campionarie; percentili campionari e box plot. La disuguaglianza di Chebychev. Insiemi di dati bivariati e coefficiente di correlazione campionaria.

ELEMENTI DI PROBABILITA'

Spazi degli esiti discreti e continui, eventi. Richiami di teoria degli insiemi e l'algebra degli eventi. Assiomi della probabilità. Spazi di probabilità finiti. Elementi di calcolo combinatorio Probabilità condizionata. Eventi indipendenti. Fattorizzazione di un evento e formula di Bayes.

VARIABILI ALEATORIE

Variabili aleatorie discrete e continue. Variabili aleatorie indipendenti. Valore atteso e sue proprietà. Varianza. Covarianza

MODELLI DI VARIABILI ALEATORIE

Variabili aleatorie di Bernoulli e binomiali. Variabili aleatorie di Poisson. Variabili aleatorie uniformi. Variabili aleatorie normali o gaussiane. Variabili aleatorie esponenziali. Distribuzioni che derivano da quella normale: le distribuzioni chi-quadro; le distribuzioni t-student; le distribuzioni F-Fisher.

LA DISTRIBUZIONE DELLE STATISTICHE CAMPIONARIE

La media campionaria. Legge dei Grandi numeri, Il Teorema del limite centrale. La varianza campionaria. Le distribuzioni delle statistiche di popolazioni normali.

STIMA PARAMETRICA

Intervalli di confidenza per la media di una distribuzione normale con varianza nota e con varianza incognita; intervalli di confidenza per la varianza di una distribuzione normale. Intervalli di confidenza approssimati per la media di una distribuzione di Bernoulli

VERIFICA DELLE IPOTESI

Livelli di significatività. La verifica di ipotesi sulla media di una popolazione normale quando la varianza è nota e quando la varianza è incognita (il test t). Verificare se due popolazioni normali hanno la stessa media. La verifica di ipotesi sulla varianza di una popolazione normale. La verifica di ipotesi su una popolazione di Bernoulli.

Testi di riferimento

Sheldon M. Ross, PROBABILITA' E STATISTICA per l'ingegneria e le scienze" 2a edizione, APOGEO, 2008.

D. Bertacchi, M. Bramanti, G. Guerra, Esercizi di Calcolo delle Probabilità e

Tipo testo

Testo

Statistica, Ed progetto Leonardo, Bologna.

Franco Crivellari, Analisi statistica dei dati con R, Apogeo, 2006.

Obiettivi formativi

Conoscere elementi fondamentali di statistica descrittiva. Comprendere e saper utilizzare i concetti di evento aleatorio, probabilità e variabile aleatoria. Saper riconoscere ed utilizzare le principali distribuzioni di probabilità utilizzate ai fini statistici: Distribuzione binomiale, di Poisson, Normale, t-di Student, chi-Quadro

Apprendere e saper manipolare le principali tecniche di Statistica Inferenziale: stime puntuali, intervalli di confidenza.

Saper utilizzare i principali test di ipotesi.

Prerequisiti

Analisi Matematica, Matematica Discreta e Logica

Modalità di verifica dell'apprendimento

Esame Scritto, composto da test a risposta chiusa e esercizi

Risultati di apprendimento attesi

Capacità di leggere dati statistici, di manipolare gli elementi fondamentali del calcolo delle probabilità e di effettuare i principali test di statistica inferenziale.

Frequenza

La frequenza alle lezioni non è obbligatoria.



Testi in inglese

Tipo testo

Testo

Lingua insegnamento

Italian

Contenuti

INTRODUCTION TO STATISTICS: populations and sets of data.

DESCRIPTIVE STATISTICS: data organization, classes of data, graphical presentation, histograms, stem and leaf representation.

Parameters: mean, variance, standard deviation, range, median. Interquartile range, boxplot. Chebychev inequality. Bivariate sets of data.

PROBABILITY: sample spaces, events. Elements of set theory, Boole algebra of events. Probability axioms.

Finite sample spaces. Elements of counting techniques. Conditional probability. Independent events, factorization of a event, Bayes Theorem.

RANDOM VARIABLE: countable and continuous random variables, independent random variable. Expected value, variance, covariance.

RANDOM VARIABLES AND PROBABILITY LAWS: Bernoulli and binomial law; geometric law, Poisson law. Uniform, Normal and exponential continuous laws.

Laws obtained by means of gaussian distribution: t-distribution, chi-square distribution, Fisher distribution.

INFERENCE STATISTICS: Sample means, Law of Large Numbers, Central Limit Theorem, probability law approximations.

ESTIMATION OF PARAMETERS: confidence intervals for the mean of normal distributions with known and unknown variance; confidence intervals for the variance of normal distributions, confidence interval for the mean of Bernoulli distributions.

TESTS OF HYPOTHESIS: tests of hypothesis for the mean of normal distributions with known and unknown variance; test of hypothesis for the variance of normal distributions; test of hypothesis for the mean of Bernoulli distributions; Test of hypothesis for two means and two variances

Testi di riferimento

Sheldon M. Ross, *PROBABILITA' E STATISTICA per l'ingegneria e le scienze* 2a edizione, APOGEO, 2008.

D. Bertacchi, M. Bramanti, G. Guerra, *Esercizi di Calcolo delle Probabilità e Statistica*, Ed progetto Leonardo, Bologna.

Franco Crivellari, *Analisi statistica dei dati con R*, Apogeo, 2006.

Prerequisiti

Mathematical Analysis, Discrete Mathematics and Logic

Modalità di verifica dell'apprendimento

Written examination, consisting of tests and exercises

Tipo testo

Testo

Frequenza

Attendance at lectures is not mandatory.

Testi del Syllabus

Docente	MIGLIORE ERNESTO	Matricola: 061669
Anno offerta:	2014/2015	
Insegnamento:	MFN0598 - FISICA (Physics	
Corso di studio:) 008707 - INFORMATICA (Computer Science)	
Anno regolamento:	2013	
CFU:	6	
Settore:	FIS/01	
Tipo attività:	C - Affine/Integrativa	
Partizione studenti:	-	
Anno corso:	2	
Periodo:	Secondo Semestre	

Tipo testo	Testo
Lingua insegnamento	Italiano
Contenuti	<p>La meccanica: moto, velocità, accelerazione, le leggi della dinamica. Forza elettrostatica e campo elettrostatico. Teorema di Gauss. Lavoro, energia meccanica e forze conservative. Conservatività del campo elettrostatico. Superfici equipotenziali. Conduttori e dielettrici. Capacità elettrica di un conduttore. Condensatori. Densità di energia del campo elettrico. Correnti elettriche. Leggi di Ohm, di Kirchoff e di Joule. Circuiti RC. Il campo magnetico indipendente dal tempo. Magnetì. Moto di una carica in campo magnetico; esempi ed applicazioni. Filo percorso da corrente in campo magnetico. Campo magnetico generato da un filo percorso da corrente. Teorema di Ampere. Teorema di Gauss per il campo magnetico. Campi elettrici e magnetici variabili nel tempo. Induzione elettromagnetica. Legge di Faraday-Henry. Correnti alternate. Legge di Ampere-Maxwell. Autoinduzione. Induttanza del solenoide ideale. Densità di energia del campo magnetico. Circuiti RL. Elementi circuitali in corrente alternata. Equazioni di Maxwell e onde elettromagnetiche. Trasmissione dei segnali (cenni)</p> <p>I programmi dettagliati relativi ai due moduli si trovano sulle pagine dei docenti. È possibile consultare anche il registro delle lezioni dell'A.A. 2013-2014. Il programma corretto in base agli argomenti effettivamente svolti sarà disponibile a fine corso. È opportuno l'utilizzo di un testo, scelto tra quelli indicati dal docente.</p>
Testi di riferimento	<p>SERWAY e JEWETT, Principi di Fisica, Ed. Edises. SERWAY, Fisica, per Scienze ed Ingegneria, vol II, Ed. Edises</p> <p>R. WOLFSON, Fisica, vol.2 - Elettromagnetismo, ottica e fisica moderna, editore Pearson</p> <p>GETTYS, KELLER, SKOVE, Fisica 2. Elettromagnetismo, Onde, Ottica Editrice McGraw-Hill RESNICK HALLIDAY, Fisica, Casa Editrice Ambrosiana. HALLIDAY, Fondamenti di Fisica, vol II, Casa Editrice Ambrosiana</p>
Obiettivi formativi	<p>Il corso si propone di introdurre alla conoscenza della basi della meccanica, delle principali proprietà del campo elettrico e del campo magnetico, con cenni al comportamento della materia soggetta a tali campi; introdurre alla conoscenza del comportamento degli elementi di un circuito in corrente continua ed in corrente alternata; introdurre alla conoscenza delle principali proprietà delle onde elettromagnetiche, con cenni ai problemi connessi alla trasmissione di tali onde.</p>
Prerequisiti	<p>Conoscenza delle nozioni fondamentali del calcolo differenziale e della trigonometria.</p>
Propedeutico a	<p>Nessun corso.</p>
Modalità di verifica dell'apprendimento	<p>Voto finale (esame scritto).</p>
Risultati di apprendimento attesi	<p>Comprensione, corredata della necessaria trattazione matematica, delle leggi fondamentali della fisica in particolare dell'elettricità e del magnetismo; capacità di risolvere analiticamente semplici problemi relativi a questa materia.</p>

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

Consigliata ma non obbligatoria.



Testi in inglese

Tipo testo

Testo

Lingua insegnamento

Italian

Contenuti

Mechanics: motion, velocity, acceleration, laws of motion. Electrostatic force and electrostatic field. Gauss' law. Work, energy and conservative forces. The electrostatic field as an example of conservative forces. Equipotential surfaces. Conductors and dielectrics. Capacity of a conductor. Capacitors. Density of energy associated to the electric field. Electric currents. Ohm's law, Kirchoff's laws and Joule's law. RC circuits. Time independent magnetic field. Magnets. Motion of a charge in a magnetic field; examples and applications. Rectilinear current in a magnetic field. Magnetic field generated by a rectilinear current. Ampere's law. Gauss' law for the magnetic field. Time dependent electric and magnetic fields. Electromagnetic induction. Faraday-Henry law. Alternating currents. Ampere-Maxwell law. Self-induction. Inductance of an ideal rectilinear solenoid. Density of energy associated to the magnetic field. RL circuits. Circuit elements in alternating current. Maxwell's equations and electromagnetic waves. Transmission of signals.

The detailed program can be found on the pages of the teachers. The register of the lessons for A.A. 2013-2014 is available as well. The final program based on the arguments actually discussed will be available at the end of the course. Students should consider the use of a text, chosen among those indicated by the teacher.

Obiettivi formativi

The course aims to introduce: the foundations of mechanics, the main properties of the electric field and of the magnetic field, including a short description of the behavior of matter subject to these fields; the behavior of the elements of a circuit in DC and AC current; the main properties of electromagnetic waves, outlining the problems associated with the transmission of the corresponding waves.

Prerequisiti

Knowledge of the basics of differential calculus and trigonometry.

Propedeutico a

None.

Modalità di verifica dell'apprendimento

Final grade (written examination).

Risultati di apprendimento attesi

Understanding, within the proper mathematical framework, of the fundamental laws of physics in particular of the electricity and of the magnetism; ability to solve analytically simple problems related to this subject.

Frequenza

Recommended but not required.

Testi del Syllabus

Docente	ARDISSONO LILIANA	Matricola: 013903
Anno offerta:	2014/2015	
Insegnamento:	MFN0608 - INTERAZIONE UOMO MACCHINA E TECNOLOGIE WEB (Human-computer Interaction and Web Technologies)	
Corso di studio:	008707 - INFORMATICA (Computer Science)	
Anno regolamento:	2012	
CFU:	12	
Settore:	INF/01	
Tipo attività:	B - Caratterizzante	
Partizione studenti:	-	
Anno corso:	3	
Periodo:	Primo Semestre	



Testi in italiano

Tipo testo

Testo

Lingua insegnamento

Italiano.

Contenuti

Parte Ia - Human-computer interaction (HCI)

Evoluzione e definizione di HCI

Il fattore umano: visione, percezione (gestalt e affordance), attenzione e memoria, modelli mentali, metafore, il modello di Shneiderman e il modello di Norman

Disegno di interazioni: user-centered design. prototipazione, linee guida, elementi di tipografia elettronica e di layout

Tecniche di valutazione

Accessibilita`

Parte Ib - Programmazione di device mobili.

Introduzione alla programmazione per mobile.

La piattaforma Android e sua architettura.

Processi e applicazioni in Android.

Progettazione di una interfaccia utente in maniera programmatica e dichiarativa.

Utilizzo dell'emulatore e di Eclipse.

Esempio di sviluppo di una semplice app per Android.

Parte II - Architetture delle applicazioni Web e tecnologie di supporto

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

Il primo livello (client dell'applicazione):

Scripting lato client (JavaScript e AJAX). Raccolta dati (via HTML form) e interazione con il web server.

Il terzo livello (livello dei dati)

Accesso a database relazionali: driver ODBC (Open Database Connection); Java Database Connectivity (JDBC).

XML: rappresentazione di informazioni (XML Schema e DTD); tecnologie di manipolazione di documenti XML (XPath).

Il secondo livello (logica applicativa)

Progettazione e sviluppo di applicazioni basate su pagine Web dinamiche (Java Server Pages e Servlet Java) e Java Beans per l'accesso a database.

Il Pattern Model View Controller per le applicazioni Web.

Progettazione e sviluppo di applicazioni Web a 3 livelli basate sul pattern MVC; sperimentazione con il framework MVC Spring.

Sperimentazione con PHP: sviluppo di applicazioni web con accesso a DB relazionale.

Testi di riferimento

Libri di Testo:

Polillo, R. - FACILE DA USARE, Edizioni Apogeo, 2010

Programmazione Web lato Server - edizione aggiornata, di V. Della Mea, L. Di Gaspero, I. Scagnetto, Apogeo, 2011

Android Training. <http://developer.android.com/training/index.html>

Altri testi (per gli interessati):

Programmazione Java - tecniche avanzate, di Deitel e Deitel. Ed. Pearson

Tipo testo

Testo

- Prentice Hall

Principi di Web design, di Joe Sklar, Apogeo

Ed Burnette. Hello, Android: Introducing Google's Mobile Development Platform. The Pragmatic Programmers, 2010.

Reto Meie. Professional Android 4 Application Development. Wiley, 2012.

Obiettivi formativi

Gli obiettivi del corso sono l'acquisizione di competenze sia teoriche sia pratiche nel disegno corretto di interazioni uomo-macchina e nella progettazione di applicazioni Web interattive, accessibili da terminali desktop e mobili (come per esempio gli smart phone).

Per quanto riguarda l'interazione uomo-macchina, verranno acquisite competenze approfondite su fondazioni di disegno user-centered ed aspetti di accessibilità alle applicazioni da parte di persone con disabilità.

Per quanto riguarda gli aspetti architettonici e tecnologici, il corso si focalizzerà sulle tecnologie Web in ambiente Java e PHP. In particolare, sul server-side programming, ossia nella progettazione e nello sviluppo di applicazioni basate su architetture modulari per l'accesso a sorgenti dati eterogenee via Web. Inoltre, il corso tratterà la tecnologia XML, data la sua importanza nella condivisione di informazioni in Internet e nella gestione di interoperabilità tra applicazioni.

Per quanto riguarda la programmazione di device mobili, verrà introdotta la piattaforma Android e verranno acquisite le competenze per la realizzazione di app per tale ambiente.

Prerequisiti

Si richiede una buona conoscenza delle basi di dati, dell'analisi e della progettazione object-oriented e dei fondamenti della programmazione distribuita.

Propedeutico a

-

Note

Il materiale usato durante le lezioni in aula e in laboratorio, i testi degli esercizi e la documentazione sono rese disponibili tramite il supporto online ai corsi <http://informatica.i-learn.unito.it/>

Modalità di verifica dell'apprendimento

L'esame è composto da una prova scritta e da una prova di laboratorio. Le due prove possono essere sostenute in qualsiasi ordine. La validità delle prove parziali è limitata al periodo che intercorre tra il loro superamento e l'inizio dello stesso corso nell'anno accademico successivo.

Il voto ottenuto durante un test scritto decade se lo studente partecipa ad un altro test scritto e consegna il suo elaborato.

Prova scritta.

Per la verifica dei contenuti della "Parte I" e "Parte II (teoria)".

Durante la prova è proibito comunicare con altre persone, presenti in aula o fuori. Inoltre, non si può portare alcun tipo di materiale didattico (appunti, libri, dispense, etc.) ed è vietato usare computer, telefonini o simili.

Come da regolamento di Ateneo, ogni studente può sostenere un numero massimo di tre prove scritte durante l'Anno Accademico (cioè, consegnare il proprio elaborato tre volte - se ci si presenta allo scritto ma ci si ritira, non viene contata la partecipazione).

Prova di laboratorio.

Tipo testo

Testo

Per la verifica dei contenuti della "Parte Ib" e "Parte II (laboratorio)".

La verifica sarà svolta tramite discussione al computer. Le soluzioni agli assegnamenti per le due parti potranno essere discusse in momenti separati.

Calcolo del voto finale di esame.

Ogni parte contribuirà equamente al voto finale. Le diverse parti di esame hanno pari peso nella definizione del voto finale.

Quando si superano entrambe le prove, è necessario registrare il voto finale entro i limiti imposti dal Regolamento di Ateneo.

Risultati di apprendimento attesi

Al termine del corso, lo studente ha acquisito le competenze necessarie per il disegno di interazioni con un alto grado di usabilità ed accessibilità e per lo sviluppo di servizi utilizzati nell'ambito di organizzazioni aziendali (come agenzie di viaggio, gestione ordini e fatture, etc.) e disponibili su intranet o internet, in ambiente Java e PHP.

Frequenza

La frequenza non è obbligatoria ma è fortemente consigliata per garantire continuità nell'apprendimento.



Testi in inglese

Tipo testo

Testo

Lingua insegnamento

Italian.

Contenuti

Part Ia - Human-computer interaction (HCI)

HCI - definition and historical perspective

Human factors: vision, perception (gestalt and affordance), attention and memory, mental models, metaphors, Shneiderman's model, Norman's model

Interaction design: user-centered design. prototyping, guidelines, electronic typography and layout design

Evaluation techniques

Accessibility

Part Ib - Mobile Programming

Introduction to mobile programming.

The Android platform and its architecture.

Processes and applications in Android.

Programming user interface in Android: pragmatically and declaratively.

Eclipse and the Android emulator.

Example of the development of a simple app for Android.

Part II - Web-based application architecture and supporting technologies

Architectures of Web applications: Web browser and Web server; 3-tier architectures.

The first level (application client):

Client-side scripting: JavaScript and AJAX. Data collection (via HTML form) and interaction with web server.

The third level (business data):

Accessing relational databases: ODBC driver (Open Database Connection); Java Database Connectivity (JDBC).

XML: information representation (XML Schema and DTD); document manipulation techniques (XPath).

The second level (business logic):

Designing and developing applications based on dynamic pages (Java Server Pages and Java Servlets) and Java Beans for database access.

The Model View Controller pattern for web applications.

Design and development of 3-tier web applications based on MVC; experiments with MVC Spring framework.

Experiments with PHP: development of web applications with relational database access.

Testi di riferimento

Book:

Polillo, R. - FACILE DA USARE, Edizioni Apogeo, 2010

Programmazione Web lato Server - edizione aggiornata, by V. Della Mea, L. Di Gaspero, I. Scagnetto, Apogeo, 2011

Android Training. <http://developer.android.com/training/index.html>

Other references (for those who are interested in further reading):

Programmazione Java - tecniche avanzate, by Deitel e Deitel. Ed. Pearson - Prentice Hall

Tipo testo

Testo

Principi di Web design, di Joe Sklar, Apogeo
Ed Burnette. Hello, Android: Introducing Google's Mobile Development Platform. The Pragmatic Programmers, 2010.
Reto Meie. Professional Android 4 Application Development. Wiley, 2012.

Obiettivi formativi

This course provides the student with theoretical and practical competence in the appropriate design of human-computer interaction and in the design of Web-based interactive applications, which are accessible both from desktops and mobile devices (e.g., smart phones).

For human-interaction, the student will acquire in-depth competence on the foundations of user-centered design and on accessibility for disabled people.

With respect to architectural and technological aspects, the course will focus on Web technologies in Java and PHP environments, and, specifically, on server-side programming, i.e., in the design and development of applications based on modular architectures for accessing heterogeneous data sources via Web. In addition, the course will present XML, a technology which is the foundation of information sharing in Internet and of the management of interoperable applications.

Finally, as far as mobile device programming is concerned, the Android platform will be presented and the student will acquire skills for developing apps in that environment.

Prerequisiti

Background on databases, Object Oriented analysis and design, fundamentals of distributed programming.

Propedeutico a

-

Note

documentation used during lectures and lab practice and the text of the exercises will be available through the on-line support platform <http://informatica.i-learn.unito.it/>

Modalità di verifica dell'apprendimento

The exam is composed of a written test and a lab test. The two parts can be taken in any order. Results are only valid until the beginning of the course in the next academic year. The participation to a written test cancels all previous results of the same test.

Written test.

This is for the "Part Ia" and "Part II (theory)".

During the written examination, the use of notes, books, computers, smart phones, etc. is forbidden. It is also forbidden to communicate in any way with other students.

Each student can be evaluated at most three times during the Academic Year (i.e., no more than three corrections of a test).

Laboratory test.

This is for the "Part Ib" and "Part II (lab)".

The verification is done with the help of a computer. The solutions of homework for "Part Ib" and "Part II (lab)" can be taken in at different times.

Final grade.

Each of the four parts contributes equally to the final grade. It is necessary to record the final grade according the rule of

Tipo testo**Testo**

Ateneo.

Risultati di apprendimento attesi

At the end of this course the student has acquired the necessary skills for the development of highly-usable and accessible applications and of Web-based services for enterprises (e.g., travel agents, etc.), developed in Java and PHP.

Frequenza

Attendance is strongly recommended especially for the lab part of the course. This will allow students to be continuously supported in their practice and learning process.

Testi del Syllabus

Docente	RADICIONI DANIELE PAOLO	Matricola: 033604
Anno offerta:	2014/2015	
Insegnamento:	MFN0590 - LINGUA INGLESE I (English I)	
Corso di studio:	008707 - INFORMATICA (Computer Science)	
Anno regolamento:	2014	
CFU:	3	
Settore:	L-LIN/12	
Tipo attività:	E - Lingua/Prova Finale	
Partizione studenti:	-	
Anno corso:	1	
Periodo:	Annualità Singola	

Testi in italiano

Tipo testo	Testo
Lingua insegnamento	Inglese
Contenuti	<ul style="list-style-type: none">- Grammatica di base- Lessico- Pratica
Testi di riferimento	<ul style="list-style-type: none">- New English File Elementary, Oxford University Press- Essential English Grammar in Use (B1), by Raymond Murphy and Lelio Pallini- English Grammar in Use (B2), by Raymond Murphy- Discussions A-Z, Cambridge University Press
Obiettivi formativi	Corso di base di Inglese orientato alla comprensione dei testi e alla grammatica di base.
Prerequisiti	Nessuno
Propedeutico a	Propedeutico a tutti i corsi del Corso di Laurea, in cui è altamente probabile imbattersi in manuali e materiale vario in lingua inglese.
Note	Il CCS in Informatica riconosce le certificazioni PET e/o FIRST per Lingua Inglese I. A tale fine è possibile compilare e consegnare l'apposito modulo APU in Segreteria Studenti.
Modalità di verifica dell'apprendimento	Il test finale è svolto nei laboratori informatici mediante il sistema SET. La prova di Inglese I per informatica è articolata in due parti, A e B. La parte A (durata 60 min.) deve essere necessariamente superata per poter accedere alla parte B. Il superamento della parte B (durata 30 min.) completa la prova. La valutazione è automatica e comunicata immediatamente dal sistema allo studente. In ogni appello si svolgono entrambe le parti, in successione; gli studenti che hanno già superato la prima parte in un appello precedente devono dare solo la seconda.
Risultati di apprendimento attesi	Al termine del corso lo studente deve essere in grado di leggere e completare semplici frasi di uso comune.
Frequenza	Fortemente consigliata per quanti non hanno studiato inglese alle superiori.

Testi in inglese

Tipo testo	Testo
Lingua insegnamento	English
Contenuti	<ul style="list-style-type: none">- Basic grammar- lexicon- Conversation and practice
Testi di riferimento	<ul style="list-style-type: none">- New English File Elementary, Oxford University Press- Essential English Grammar in Use (B1), by Raymond Murphy and Lelio Pallini- English Grammar in Use (B2), by Raymond Murphy- Discussions A-Z, Cambridge University Press
Obiettivi formativi	Introductory level course, aimed at texts comprehension and at getting familiar with basic English grammar.
Prerequisiti	None
Propedeutico a	In principle all courses (and collecting materials for the final dissertation, as well) require to get familiar (at least) with technical English.
Note	The CCS in Computer Science accepts PET and FIRST certifications in place of the English I exam. The APU module is made available by the Students' administrative office.
Modalità di verifica dell'apprendimento	Exams will take place in the Department labs. They are composed of two parts, A and B. The part A (lasting 60 minutes) must be passed in order to gain access to the part B (lasting 30 mins.). The evaluation of the test is performed automatically through an ad hoc software. The two parts take place in sequence, so that students having already passed part A (in a previous exam) only need to repeat the part B.
Risultati di apprendimento attesi	At the end of the course, it is expected that students are able to read and complete (on a 'fill in the blanks' basis) simple sentences of common use.
Frequenza	Attending the lessons is strongly recommended for those who did not study English before.

Testi del Syllabus

Docente	DEZANI MARIANGIOLA	Matricola: 010910
Anno offerta:	2014/2015	
Insegnamento:	MFN0610 - LINGUAGGI E PARADIGMI DI PROGRAMMAZIONE (Programming languages and paradigms)	
Corso di studio:	008707 - INFORMATICA (Computer Science)	
Anno regolamento:	2012	
CFU:	9	
Settore:	INF/01	
Tipo attività:	B - Caratterizzante	
Partizione studenti:	-	
Anno corso:	3	
Periodo:	Primo Semestre	



Testi in italiano

Tipo testo	Testo
Lingua insegnamento	Italiano
Contenuti	<p>Nel seguente elenco di argomenti non viene fatta distinzione tra argomenti svolti in aula ed argomenti svolti in laboratorio, che sono peraltro strettamente connessi.</p> <p>Introduzione ai paradigmi di programmazione: programmazione imperativa, orientata agli oggetti, logica e funzionale;</p> <p>Calcolo come riscrittura: le basi dell'esecuzione dei programmi funzionali;</p> <p>Espressioni e loro tipi. Tipi di base;</p> <p>Progettazione di programmi funzionali. Tecniche di ricorsione;</p> <p>Liste e funzioni del prim'ordine su liste;</p> <p>Ragionare su programmi funzionali: tecniche di induzione;</p> <p>Testing di programmi Haskell con QuickCheck;</p> <p>L'idea di astrazione funzionale. Funzioni di ordine superiore e pattern di calcolo.</p> <p>Programmazione con funzioni di ordine superiore;</p> <p>alutazione Lazy. Strutture dati potenzialmente infinite.</p>
Testi di riferimento	<p>Simon Thompson, "Haskell: the craft of functional programming", Terza edizione, Addison-Wesley, 2011</p> <p>Materiale aggiuntivo rispetto al testo (codice, note) verrà reso disponibile durante lo svolgimento delle lezioni.</p>
Obiettivi formativi	<p>Apprendimento dei principi su cui si basano i principali paradigmi di programmazione, in particolare il paradigma funzionale, e la loro realizzazione in linguaggi di programmazione esistenti. Il corso verte principalmente sul linguaggio di programmazione funzionale puro Haskell.</p>
Prerequisiti	<p>Nozioni di programmazione, algoritmi e linguaggi formali.</p>
Modalità di verifica dell'apprendimento	<p>Esame orale con esercizi e discussione dei progetti di laboratorio</p>

Testi in inglese

Tipo testo

Testo

Lingua insegnamento

Italian

Contenuti

In the following list of topics no distinction is made between theory classes and lab sessions.

Introduction to programming paradigms: imperative, object-oriented, logic, and functional

programming;

Computation as rewriting: executing functional programs;

Expressions and their types. Basic types;

Designing functional programs. Recursion techniques.

Lists and first-order functions over lists;

Reasoning about functional programs: induction techniques.

Testing Haskell programs using QuickCheck;

Functional abstraction. Higher-order functions and computation patterns;

Trees and algebraic data types;

Lazy evaluation. Potentially infinite data structures.

Testi di riferimento

Simon Thompson, "Haskell: the craft of functional programming", Terza edizione, Addison-Wesley, 2011

Additional didactic material (notes, code) will be available during the course.

Obiettivi formativi

Learning of the most important programming paradigms, in particular the functional one, and of their realization in the existing programming languages.

Prerequisiti

Notions of programming, algorithms and formal languages

Modalità di verifica dell'apprendimento

Oral examination and discussion of the projects developed in laboratory.

Testi del Syllabus

Docente	ZACCHI MADDALENA	Matricola: 011333
Anno offerta:	2014/2015	
Insegnamento:	MFN0603 - LINGUAGGI FORMALI E TRADUTTORI (Formal Languages and Compilers)	
Corso di studio:	008707 - 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

Concetti centrali della teoria dei linguaggi: 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. 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. Laboratorio: uso di strumenti per la generazione automatica di parsificatori. Sviluppo di programmi Java per l'interpretazione e la compilazione di un semplice linguaggio di programmazione imperativo.

Testi di riferimento

J. E. Hopcroft, R. Motwani, J. D. Ullman, "Automati, 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 che sono considerate acquisite nei corsi successivi.

Tipo testo

Modalità di verifica dell'apprendimento

Testo

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.
- una prova di laboratorio: realizzazione di un interprete/compiler per un semplice linguaggio definito ad hoc.

Di norma la prova di laboratorio viene fissata ad una settimana di distanza dalla prova scritta.

Risultati di apprendimento attesi

Conoscenza dei principi fondamentali della sintassi dei linguaggi di programmazione e delle principali metodologie di parsificazione e traduzione. Conoscenza e padronanza 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. Equivalence of different types of finite state automata and their applications.
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 recursive descent parsing.
Intermediate code generation: three-address code. Translation of arithmetic expressions, Boolean expressions and translation of statements.
Laboratory: use of tools for the automatic generation of parsers. Development of Java programs for the interpretation and the compilation of a simple imperative programming language.

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 taken for granted in the subsequent courses.

Tipo testo

Modalità di verifica dell'apprendimento

Testo

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.
- A laboratory test: development of an interpreter / compiler for a simple language defined ad hoc.

As a rule, the laboratory test is fixed to a week after the written tests.

Risultati di apprendimento attesi

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

Testi del Syllabus

Docente **ANDRETTA ALESSANDRO** Matricola: **013650**

Anno offerta: **2014/2015**

Insegnamento: **MFN0578 - MATEMATICA DISCRETA E LOGICA (Discrete Mathematics and Logic)**

Corso di studio: **008707 - INFORMATICA (Computer Science)**

Anno regolamento: **2014**

CFU:

Tipo attività: -

Partizione studenti: -

Anno corso: **1**

Periodo: **Primo Semestre**

Testi in italiano

Tipo testo	Testo
Lingua insegnamento	Italiano
Contenuti	Insiemi e relazioni d'equivalenza. Strutture algebriche: semigrupp monoidi, anelli, campi. Aritmetica modulare. Calcolo combinatorio. 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 concept di base dell'algebra e della logica matematica
Prerequisiti	Una ragionevole maturità matematica
Propedeutico a	nessun corso
Modalità di verifica dell'apprendimento	Esame 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
Frequenza	Facoltativa, ma consigliata



Testi in inglese

Tipo testo	Testo
Lingua insegnamento	Italian
Contenuti	Sets and equivalence relations. Algebraic structures: semigroups, monoids, rings, fields. Modular arithmetic. Combinatorial calculus. 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 basics of algebra and of mathematical 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
Frequenza	Optional, but encouraged

Testi del Syllabus

Docente	RONCHI DELLA ROCCA SIMONETTA	Matricola: 011223
Anno offerta:	2014/2015	
Insegnamento:	MFN0633 - METODI FORMALI DELL'INFORMATICA (Formal Methods in Computer Science)	
Corso di studio:	008707 - INFORMATICA (Computer Science)	
Anno regolamento:	2012	
CFU:	9	
Settore:	INF/01	
Tipo attività:	B - Caratterizzante	
Partizione studenti:	-	
Anno corso:	3	
Periodo:	Secondo Semestre	

Tipo testo	Testo
Lingua insegnamento	Italiano
Contenuti	<p>Teoria della calcolabilità:</p> <ul style="list-style-type: none">- problemi decidibili e indecidibili;- modelli di computazione (macchine di Turing, linguaggio while, funzioni di Kleene);- teorema dell'interprete;- tesi di Church-Turing. <p>Teoria della complessità:</p> <ul style="list-style-type: none">- complessità temporale e spaziale;- classi di complessità;- le classi P e NP;- problemi NP-completi. <p>Semantica formale:</p> <ul style="list-style-type: none">- semantica operativa di un semplice linguaggio di programmazione imperativo;- ordini parziali, funzioni continue e teorema del punto fisso;- cenni di semantica denotazionale.
Testi di riferimento	<ul style="list-style-type: none">- Kfoury, Moll, Arbib: "A Programming Approach to Computability", Springer-Verlag, 1982 (Il libro è fuori pubblicazione, ma si possono trovare le fotocopie - informazioni presso la biblioteca del Dipartimento di Informatica).- M. Sipser, Introduction to the Theory of Computation, Second edition Thomson Course Technology, Boston 2006- C. Toffalori et alii, Teoria della calcolabilità e della complessità, McGraw-Hill 2005- Winskel: "La semantica formale dei linguaggi di programmazione", The MIT Press, edizione italiana UTET, 1999
Obiettivi formativi	Scopo del corso è fornire agli studenti delle nozioni di base per poter ragionare criticamente sul significato dei programmi, sulla relazione tra programmi e funzioni, sulla complessità dei problemi.
Prerequisiti	Si prevede che lo studente abbia acquisito le nozioni base di matematica discreta e logica (teoria elementare degli insiemi, linguaggio della logica e nozione di prova, induzione matematica). Nell'ambito informatico, deve conoscere le nozioni di algoritmo e programma, e conoscere almeno un linguaggio di programmazione. Più precisamente, si richiede che lo studente abbia seguito i corsi di Matematica Discreta e Logica, Programmazione 1 e 2, Algoritmi e Strutture Dati,
Modalità di verifica dell'apprendimento	L'esame consiste in un esame orale. Verrà anche tenuto conto della partecipazione degli studenti agli esercizi in classe.
Risultati di apprendimento attesi	Gli studenti debbono saper ragionare criticamente sul significato dei programmi, sulla relazione tra programmi e funzioni, sulla complessità dei problemi.
Frequenza	La frequenza non è obbligatoria, ma caldamente consigliata.



Testi in inglese

Tipo testo

Testo

Lingua insegnamento

Italian

Contenuti

Computability:
- decidable and undecidable problems;
- models of computation (Turing machines, while-language, Kleene functions);
- universal function theorem;
- thesis of Church-Turing.
Complexity:
- complexity in time and space;
complexity classes;
- P and NP classes;
- NP-complete problems.
Formal semantics:
- operational semantics of a simple imperative language;
- partial order, continuous functions and fixed point theorem;
- introduction to denotational semantics.

Testi di riferimento

- Kfoury, Moll, Arbib: "A Programming Approach to Computability", Springer-Verlag, 1982 (The book is no more edited, but copies and information are at the library of the Computer Science Department).
- M. Sipser, Introduction to the Theory of Computation, Second edition Thomson Course Technology, Boston 2006
- C. Toffalori et alii, Teoria della calcolabilita' e della complessita', McGraw-Hill 2005
- Winskel: "La semantica formale dei linguaggi di programmazione", The MIT Press, edizione italiana UTET, 1999

Obiettivi formativi

Aim of this course id to give to the students the basic notions to reason in a critical way on the meaning of programs, on the relation between programs and functions, on the complexity of problems.

Prerequisiti

Students are assumed to know the fundamental notions about discrete mathematics and logics (elementary set theory, first order logic, logical proofs, mathematical induction). From a computer science point of view, they need to know the notions of algorithm and program, to be able to use at least one programming language. More precisely, we assume the students did attend the courses of Discrete Mthematics and Logic, Programming 1 and 2, Algorithms and Data Structures.

Modalità di verifica dell'apprendimento

The verification will be through an oral examination. The participation to the practical exercises will be taken into account.

Risultati di apprendimento attesi

Students need to be able to reason in a critical way on the meaning of programs, on the relation between programs and functions, on the complexity of problems.

Frequenza

Students are suggested to follow the lessons.

Testi del Syllabus

Docente

ROVERSI LUCA

Matricola: **013941**

Anno offerta:

2014/2015

Insegnamento:

MFN0582 - PROGRAMMAZIONE I (Programming I)

Corso di studio:

008707 - INFORMATICA (Computer Science)

Anno regolamento:

2014

CFU:

9

Settore:

INF/01

Tipo attività:

A - Base

Partizione studenti:

-

Anno corso:

1

Periodo:

Primo Semestre



Testi in italiano

Tipo testo

Testo

Lingua insegnamento

Italiano

Contenuti

- * Struttura di base di un calcolatore.
- * Linguaggi di programmazione, compilatori e interpreti.
- * Algoritmi iterativi e ricorsivi.
- * Linguaggio di riferimento: variabili, tipi di dato fondamentali e array, assegnazione e controllo del flusso, procedure e funzioni con parametri.
- * Correttezza, terminazione e costo degli algoritmi.

Testi di riferimento

Walter Savitch
"Programmazione di base e avanzata con Java"
Pearson --- ISBN 9788865181904

Obiettivi formativi

Il corso ha lo scopo di fornire i concetti di base della programmazione imperativa, appoggiandosi ad un linguaggio di programmazione di riferimento.

Prerequisiti

Il corso non presuppone alcuna conoscenza di programmazione o piu` in generale di informatica, al di la` della semplice capacita` di usare un calcolatore con sistema a finestre (gestione di files, elaborazione di testi).

Per quanto riguarda la matematica, il modulo presuppone solo conoscenze fondamentali, quali i concetti di numero (naturale, intero, razionale, reale), di funzione, le quattro operazioni, elevamento a potenza, radice, esponenziale, logaritmo, il piano cartesiano, il calcolo letterale elementare. Si presuppongono invece capacita` di ragionamento logico, di astrazione e di risoluzione non meccanica di problemi, accompagnate da una buona padronanza della lingua madre.

Propedeutico a

essenzialmente tutti i corsi INF/01, pur non essendo obbligatorio aver sostenuto l'esame per seguire corsi successivi e sostenere altri esami INF/01.

Note

- Come supporto on-line alla didattica potranno essere disponibili:
- * Dispense sui principi della programmazione
 - * Puntatori a documentazione on-line
 - * Software

Modalità di verifica dell'apprendimento

- * Codifica di algoritmi iterativi e ricorsivi.
- * Esercizi in grado di verificare la preparazione sugli aspetti teorici della programmazione (correttezza, terminazione, simulazione della gestione della memoria).

Risultati di apprendimento attesi

Capacità di: (i) formalizzare la soluzione a problemi computazionali di base per mezzo di costrutti linguistici iterativi e ricorsivi, (ii) tradurre la formalizzazione in un linguaggio di programmazione di riferimento, (iii) valutare, anche per mezzo di semplici dimostrazioni formali, la correttezza e la terminazione della soluzione proposta, (iv) valutare l'efficienza della soluzione proposta.

Frequenza

Non obbligatoria, ma vivamente consigliata.



Testi in inglese

Tipo testo	Testo
Lingua insegnamento	Italian
Contenuti	<ul style="list-style-type: none">* Basic structure of a computer* Programming languages, compilers and interpreters* Iterative and recursive algorithms* Language of reference: variables, basic data types and arrays, assignment and control flow, procedures and function with parameters* correctness, finiteness and cost of algorithms
Testi di riferimento	Walter Savitch "Programmazione di base e avanzata con Java" Pearson --- ISBN 9788865181904
Obiettivi formativi	The course is about basic concepts relative to imperative programming also by means of a programming language of reference.
Prerequisiti	<p>No prerequisites exist neither about programming nor about computer science. Some basic skill on how using standard graphic interfaces to access a file system may help.</p> <p>Basic mathematical concepts are assumed: number (natural, integer, rational, real), function, basic operations on numbers, exponentiation, logarithm, use of the cartesian plane, elementary symbolic calculus. Attitude and interest to structured reasoning, abstraction and problems solving abilities and a good mastery of mother tongue are desirable.</p>
Propedeutico a	essentially every of the courses INF/01, even though it will not be mandatory passing the exam of Programmazione 1 to attend lessons and exams of succeeding courses INF/01.
Note	Possible on-line support will be: <ul style="list-style-type: none">* Notes about programming* Pointers to further on-line documentation* Software
Modalità di verifica dell'apprendimento	<ul style="list-style-type: none">* Coding of iterative and recursive algorithms* Exercises to assess skills about theoretical aspects of programming (correctness, termination, simulation of a memory model.)
Risultati di apprendimento attesi	Abilities to: (i) formalize solutions to basic computational problems by means of iterative and recursive constructs, (ii) translate the formalization into a programming language of reference, (iii) assess, possibly by simple formal proofs, the correctness and the termination of the given solution, (iv) evaluate the efficiency of the given solution.
Frequenza	It is not mandatory, but strongly encouraged.

Testi del Syllabus

Docente	ARDISSONO LILIANA	Matricola: 013903
Anno offerta:	2014/2015	
Insegnamento:	MFN0585 - PROGRAMMAZIONE II (Programming II and Laboratory)	
Corso di studio:	008707 - INFORMATICA (Computer Science)	
Anno regolamento:	2014	
CFU:	9	
Settore:	INF/01	
Tipo attività:	A - Base	
Partizione studenti:	-	
Anno corso:	1	
Periodo:	Secondo Semestre	



Testi in italiano

Tipo testo	Testo
Lingua insegnamento	Italiano
Contenuti	<p>Il corso tratta i seguenti temi, utilizzando il linguaggio Java:</p> <ul style="list-style-type: none">- Introduzione agli oggetti (stato, operazioni) tramite il modello della memoria della macchina virtuale Java.- Composizione di oggetti.- Le liste concatenate con ripasso sulla ricorsione, affrontata a Programmazione I.- Le interfacce Java e il polimorfismo.- Alcune strutture dati: pila, coda, lista ordinata.- Introduzione all'ereditarietà.- Overriding e binding dinamico.- La keyword super nei metodi.- Le chiamate this() e super() nei costruttori.- Upcast, downcast; introduzione ai tipi generici.- Overriding e overloading.- Principi di Object-Oriented Design- Trattamento delle eccezioni.- La struttura dati albero binario.
Testi di riferimento	<p>W. Savitch: Programmazione di base e avanzata con Java, 1 edizione. Pearson Education.</p> <p>Allen B. Downey. How to Think Like a Computer Scientist: Java Version. Disponibile on-line.</p>
Obiettivi formativi	<p>Il corso si propone di raffinare le capacità di programmare nel linguaggio Java apprese nel corso di Programmazione I e di introdurre i concetti di programmazione orientata agli oggetti. Si darà particolare enfasi agli aspetti di buona progettazione del software, utilizzando alcuni concetti presi a prestito dall'ingegneria del software e formalismi grafici quali UML.</p>
Prerequisiti	<p>Nozioni fondamentali di programmazione e linguaggio Java acquisite nel corso di Programmazione I.</p>
Propedeutico a	<p>Corsi INF/01 dal secondo anno in avanti, pur non essendo obbligatorio aver sostenuto l'esame per seguire corsi successivi e sostenere altri esami INF/01.</p>
Note	<p>Il materiale usato durante le lezioni in aula e in laboratorio, i testi degli esercizi e tutta la documentazione sono resi disponibili tramite il supporto on-line ai corsi http://informatica.i-learn.unito.it/</p>
Modalità di verifica dell'apprendimento	<p>L'esame si compone di una verifica di laboratorio e di una prova scritta.</p> <p>La VERIFICA DI LABORATORIO, comune a tutti, si articola in una prova al</p>

Tipo testo

Testo

calcolatore della durata di 45 minuti nella quale viene richiesto di risolvere un semplice problema di programmazione che presuppone l'aver risolto e compreso gli esercizi svolti durante i laboratori.

L'esame di laboratorio viene valutato con un punteggio da 0 a 6, dove 0 equivale a una bocciatura.

Una volta sostenuta la prova di laboratorio, il punteggio ottenuto rimarrà valido per tutti i rimanenti appelli dell'anno accademico in corso, dopodichè la prova di laboratorio dovrà essere ripetuta.

Uno studente può CONSEGNARE a un massimo di 3 esami di laboratorio durante l'anno accademico. Può però presentarsi ad ogni appello (previa registrazione), e decidere se consegnare il proprio elaborato.

La PROVA SCRITTA include domande teoriche e esercizi. Viene valutata da 0 a 26 punti e viene considerata superata se si ottengono almeno 16 punti. Non ci si può presentare all'esame scritto se non si è superata prima la verifica di laboratorio. Gli studenti possono sostenere l'esame di Programmazione II e Laboratorio anche se non hanno ancora superato l'esame di Programmazione I.

Uno studente può CONSEGNARE a un massimo di 3 esami scritti durante l'anno accademico. Può però presentarsi ad ogni appello (previa registrazione), e decidere se consegnare il proprio elaborato.

Il voto finale è ottenuto sommando il voto della parte di laboratorio con il voto dello scritto. Un voto complessivo superiore a 30 equivale a 30 e lode.

Risultati di apprendimento attesi

Conoscenza di base della programmazione e progettazione object-oriented. Maggior capacità di programmare con metodi ricorsivi e non. Conoscenza di strutture dati fondamentali (liste e alberi).

Frequenza

La frequenza non è obbligatoria ma è fortemente consigliata per permettere continuità nell'apprendimento.



Testi in inglese

Tipo testo

Testo

Lingua insegnamento

Italian

Contenuti

The course presents the following themes by means of the Java language:

- Introduction to objects (state, operations) via the Java virtual machine memory model.
- Object composition.
- Linked lists with revision on recursion, introduced in Programmazione I.
- Java interfaces and polymorphism.
- Some data structures: stack, queue, ordered list.
- Introduction to inheritance.
- Overriding and dynamic binding.
- The super keyword in methods.
- The this() and super() calls in constructors.
- Upcast, downcast; an introduction on generics.
- Overriding and overlading.
- Object-Oriented Design principles.
- Java exceptions treatment.
- The binary tree data structure.

Testi di riferimento

W. Savitch: Programmazione di base e avanzata con Java, 1st edition. Pearson Education.

Allen B. Downey. How to Think Like a Computer Scientist: Java Version. Available on-line.

Obiettivi formativi

The course aims to refining the programming skills acquired during the Programmazione I course and introducing concepts of object-oriented programming. We will put emphasis on good design of software by means of concepts borrowed from software engineering and graphical formalisms such as UML.

Prerequisiti

Basic programming skills and knowledge about the Java language provided by the Programmazione I course.

Propedeutico a

Subsequent INF/01 courses, even though passing Programmazione II exam is not mandatory for taking them.

Note

The documentation used during lectures and lab practice and the text of the exercises will be available through the on-line support platform <http://informatica.i-learn.unito.it/>

Modalità di verifica dell'apprendimento

The exam includes a laboratory and a written test.

The LABORATORY TEST consists of a programming task to be carried out using a computer. It lasts 45 minutes and the student has to solve a

Tipo testo

Testo

simple programming problem whose solution requires having completed and understood the exercises developed during the laboratory lessons of the course.

The mark obtained in the laboratory test takes values in $[0, 6]$, where 0 means that the test is insufficient.

After having passed the laboratory test, the obtained mark is valid until the last exam of the current Academic Year; afterwards it expires and the laboratory test has to be completed again.

Each student may request the evaluation of at most 3 laboratory tests during a single Academic Year. However, (s)he can participate in each exam (if registered) and decide whether (s)he wants to be evaluated or not.

The WRITTEN TEST includes theoretical questions and exercises.

The mark obtained during the written text takes values in $[0, 26]$ and any marks equal or greater than 16 are considered as sufficient. In order to participate in the written test the student must pass the laboratory test. Note that students can take the exams of this course even though they have not already passed the exam of the Programmazione I course.

Each student may request the evaluation of at most 3 written tests during a single Academic Year. However, (s)he can participate in each exam (if registered) and decide whether (s)he wants to be evaluated or not.

The final mark of the course is obtained summing the mark obtained in the laboratory test with the one obtained in the written test. A final mark which is greater than 30 is equivalent to 30 and praise.

Risultati di apprendimento attesi

Base knowledge of object-oriented programming and design. Improved programming skills (with iterative and recursive methods). Knowledge of fundamental data structures (lists and trees).

Frequenza

Attendance is strongly recommended especially for the lab part of the course. This allows students to be continuously supported in their practice and learning process.

Testi del Syllabus

Docente

BALDONI MATTEO

Matricola: **013943**

Anno offerta:

2014/2015

Insegnamento:

MFN0605 - PROGRAMMAZIONE III (Programming III)

Corso di studio:

008707 - INFORMATICA (Computer Science)

Anno regolamento:

2012

CFU:

6

Settore:

INF/01

Tipo attività:

B - Caratterizzante

Partizione studenti:

-

Anno corso:

3

Periodo:

Primo Semestre

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

Italiano.

Contenuti

Programmazione ad eventi in Java: programmare interfacce grafiche.

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

Programmazione Multithread:

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

Programmazione in rete in Java:

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

Testi di riferimento

Cay S. Horstmann e Gary Cornell.

Java 2: i fondamenti. 7a edizione.

Pearson Education Italia, 2005.

ISBN 8871922387.

Cay S. Horstmann e Gary Cornell.

Java 2: Tecniche Avanzate. 7a edizione.

Pearson Education Italia, 2005.

ISBN 8871922379.

K. Arnold, J. Gosling, D. Holmes.

Il linguaggio Java, Manuale Ufficiale.

Pearson Education Italia, 2006.

Silvia Crafa.

Programmazione concorrente e distribuita.

Libreria Internazionale Cortina, Padova, 2010.

Obiettivi formativi

Presentare aspetti importanti dei linguaggi di programmazione, facendo riferimento in particolare a Java, non trattati nei precedenti corsi di programmazione. Un primo obiettivo riguarda la tecnica di programmazione ad eventi per realizzare interfacce grafiche. Verranno poi presentati strumenti di programmazione di linguaggi ad alto livello, come thread e invocazione remota di metodi, che consentono di realizzare in modo semplice applicazioni concorrenti e distribuite in rete.

Prerequisiti

Il modulo presuppone un buona conoscenza delle tecniche di programmazione ad oggetti fornite da corsi di base di programmazione. E' inoltre richiesta una buona conoscenza dei meccanismi di base e delle problematiche della programmazione concorrente, come quelle comunemente fornite da un corso di sistemi operativi.

Sono richieste le conoscenze fornite dai corsi di:

Programmazione I, Programmazione II, Algoritmi e strutture dati, Sistemi Operativi.

Propedeutico a

Interazine Uomo Macchina, parte di programmazione android.

Tipo testo**Testo****Modalità di verifica dell'apprendimento**

Esame scritto.

Risultati di apprendimento attesi

Al termine del corso ci si aspetta che lo studente sappia realizzare applicazioni distribuite usando linguaggi ad alto livello, in particolare Java.

Frequenza

La frequenza e' fortemente consigliata.



Testi in inglese

Tipo testo

Testo

Lingua insegnamento

Italian.

Contenuti

Event driven programming: programming graphical interfaces:
- Event sources, event handlers, event-driven programming
- Organization and use of graphical interfaces in Java.
- Model-View-Control architecture.

Multithread programming:

- Executing concurrent statements.
- Threads in Java. Lifecycle of threads.
- Creation and synchronization of threads.
- Extension of the memory model to deal with threads.
- Synchronization problems and their solution with Java threads.

Network programming:

- Client-server architecture.
- Using sockets.
- Polymorphism and object transfer in Java.
- Remote method invocation.
- Modeling distributed execution of objects.

Testi di riferimento

Cay S. Horstmann e Gary Cornell.
Java 2: i fondamenti. 7a edizione.
Pearson Education Italia, 2005.
ISBN 8871922387.

Cay S. Horstmann e Gary Cornell.
Java 2: Tecniche Avanzate. 7a edizione.
Pearson Education Italia, 2005.
ISBN 8871922379.

K. Arnold, J. Gosling, D. Holmes.
Il linguaggio Java, Manuale Ufficiale.
Pearson Education Italia, 2006.

Silvia Crafa.
Programmazione concorrente e distribuita.
Libreria Internazionale Cortina, Padova, 2010.

Obiettivi formativi

The aim of this course is to introduce the student to the advanced programming topics. A first goal is to introduce to the event driven programming and the programming to the user interfaces. Moreover, the student will acquire the competence for programming in an distributed and concurrent environment by using remote method invocations and by using threads.

Prerequisiti

It is necessary to know the basic notions of object oriented programming and of concurrent programming.

Propedeutico a

Human-Computer Interaction, part of android programming.

Modalità di verifica dell'apprendimento

Written exam.

Risultati di apprendimento attesi

At the end of the course the student will have a deeper knowledge of the programming features of Java about distributed and concurrent mechanisms.

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

The attendance is highly recommended.

Testi del Syllabus

Docente	SERENO MATTEO	Matricola: 013393
Anno offerta:	2014/2015	
Insegnamento:	MFN0635 - RETI DI ELABORATORI (Computer Networks)	
Corso di studio:	008707 - INFORMATICA (Computer Science)	
Anno regolamento:	2012	
CFU:	12	
Settore:	INF/01	
Tipo attività:	B - Caratterizzante	
Partizione studenti:	-	
Periodo:	Annualità Singola	

Testi in italiano

Tipo testo	Testo
Lingua insegnamento	Italiano
Contenuti	Introduzione ad Internet ed alle reti di calcolatori Il livello di collegamento: collegamenti, reti di accesso e reti locali Reti Wireless e Mobili Il livello di Rete Il livello di Trasporto Il livello Applicazioni Come creare un'applicazione di rete: I socket Gestione della rete
Testi di riferimento	Reti di Calcolatori e Internet - Un approccio top-down (sesta edizione), J. Kurose, K. Ross, Pearson Education Altri Libri consigliati: Reti di calcolatori, di A. Tanenbaum,, Pearson, Prentice Hall Introduzione alla programmazione client-server, D. Maggiorini, Pearson, Addison Wesley Internetworking con TCP/IP. Vol. 1: Principi, protocolli e architetture, D. Comer, Pearson Prentice Hall
Obiettivi formativi	Il corso studia le reti di trasmissione, la suite di protocolli TCP/IP, e i principi che guidano la strutturazione e la progettazione di applicazioni distribuite.
Prerequisiti	Programmazione I e II, Architetture, Sistemi Operativi.
Modalità di verifica dell'apprendimento	La verifica si articolerà in due prove scritte che potranno richiedere complessivamente tra le due e le tre ore. La verifica sarà orientata ad accertare la capacità maturata dallo studente nella comprensione dei fenomeni e dei protocolli che sono alla base della trasmissione delle informazioni via cavo e via etere, nella conoscenza approfondita dei protocolli che compongono la suite TCP/IP e delle loro relazioni, e nella capacità di progettare e realizzare semplici applicazioni distribuite.
Risultati di apprendimento attesi	Comprensione dei fenomeni e dei protocolli che sono alla base della trasmissione delle informazioni via cavo e via etere; Conoscenza approfondita dei protocolli che compongono la suite TCP/IP e delle loro relazioni; Capacità di progettare e realizzare semplici applicazioni distribuite.



Testi in inglese

Tipo testo	Testo
Lingua insegnamento	Italian
Contenuti	Introduction Internet and to computer networks The Link Layer and the Local Area Networks Wireless and Mobile Networks The Network Layer The Transport Layer The Application Layer Socket programming Network Management
Testi di riferimento	Computer Networking: A Top-Down Approach, 6/e J. Kurose, K. Ross, Pearson Education Other textbooks (optional) Computer Networks, 5/E, A. S. Tanenbaum, David J. Wetherall, Prentice Hall Introduzione alla programmazione client-server, D. Maggiorini, Pearson, Addison Wesley Internetworking with TCP/IP, Vol. 1, D. Comer, Prentice Hall
Obiettivi formativi	The course focusses on Computer Networks, on Internet, and on the basic principles for the design of distributed applications
Prerequisiti	Programming, Computer Architectures, Operating Systems.
Modalità di verifica dell'apprendimento	The exam is divided into two written tests (that will require between two and three hours). These tests will verify the student knowledge about the basic phenomena and protocols concerning the information transmission (wired and wireless), the TCP/IP protocols and relationships, and the ability to design and build simple distributed applications.
Risultati di apprendimento attesi	Understanding of the phenomena and protocols that are at the basis of the transmission of information (for wired and wireless communication). Knowledge of the TCP/IP protocols and their relationships. Basic principles for the design of simple distributed applications.

Testi del Syllabus

Docente	BOTTA MARCO	Matricola: 013397
Anno offerta:	2014/2015	
Insegnamento:	MFN1362 - RETI I (Computer Networks I)	
Corso di studio:	008707 - INFORMATICA (Computer Science)	
Anno regolamento:	2012	
CFU:	6	
Settore:	INF/01	
Tipo attività:	B - Caratterizzante	
Partizione studenti:	-	
Periodo:	Primo Semestre	

Testi in italiano

Tipo testo	Testo
Lingua insegnamento	Italiano
Contenuti	Fondamenti su reti di calcolatori. Si seguirà l'approccio top-down. Il livello applicativo: suite di applicazioni TCP/IP (Web, Posta, DNS, File transfer, sistemi P2P) Il livello transport: controllo della congestione e del flusso end-to-end Il livello rete: instradamento e reti IP Il livello link: condivisione del mezzo e controllo di flusso. Il livello di comunicazione fisico: mezzi di comunicazione, modulazione, multiplexing. Reti wireless e mobilità.
Testi di riferimento	Edizione italiana: J. Kurose - K. Ross. Reti di calcolatori e internet - Un approccio top-down 4a Edizione Pearson - Addison Wesley
Obiettivi formativi	Il corso si propone di fornire agli studenti nozioni base sulle reti di calcolatori e una comprensione approfondita della suite di protocolli TCP/IP. Inoltre, attraverso l'uso di software ed esempi pratici, il corso fornisce agli studenti una comprensione concreta dei meccanismi di comunicazione tra dispositivi di rete e calcolatori.
Prerequisiti	Lo studente deve conoscere prima di seguire il corso i fondamenti della programmazione e dei sistemi operativi. Corsi fornitori: sistemi operativi, Algoritmi
Modalità di verifica dell'apprendimento	scritta
Risultati di apprendimento attesi	Il corso si propone di preparare gli studenti allo sviluppo di un progetto di tipo generico che coinvolga una tecnologia di rete relativa alla famiglia di protocolli TCP/IP. Al tempo stesso, gli studenti che superano con successo l'esame potranno inserirsi anche in gruppi di lavoro tecnici che debbano affrontare problematiche di sviluppo di applicazioni distribuite di progettazione, implementazione e/o adeguamento di protocolli di rete.
Frequenza	non obbligatoria



Testi in inglese

Tipo testo	Testo
Lingua insegnamento	Italian
Contenuti	Computer networks fundamentals, following a top down approach. Application level: overview of TCP/IP applications (web, mail, DNS, File Transfer, P2P systems). Transport level: congestion management and end-to-end transport. Network level: routing and IP networks. Link level: media access and sharing. Physical level: communication media, modulation and multiplexing. Wireless networks and mobility.
Testi di riferimento	J. Kurose - K. Ross. Computer Networking: A Top-Down approach featuring the Internet. 5/e Pearson - Addison Wesley
Obiettivi formativi	The course provides basic notion of computer networks and the protocols mainly used in TCP/IP networks. Moreover, by using software programs and examples, students can get a deeper understanding of the mechanisms that regulate devices to/from computers communications.
Prerequisiti	The student should know programming and operating systems fundamental concepts. Related courses: Sistemi Operativi, Algoritmi
Modalità di verifica dell'apprendimento	written
Risultati di apprendimento attesi	The student should be able to develop a generic project that needs a network technology of the TCP/IP family. Moreover, the learned skills allow the student to work in teams that develop distributed applications, design, implement and/or update network protocols.
Frequenza	not mandatory

Testi del Syllabus

Docente	BERGADANO FRANCESCO	Matricola: 013238
Anno offerta:	2014/2015	
Insegnamento:	MFN0636 - SICUREZZA (Computer and Network Security)	
Corso di studio:	008707 - INFORMATICA (Computer Science)	
Anno regolamento:	2012	
CFU:	6	
Settore:	INF/01	
Tipo attività:	B - Caratterizzante	
Partizione studenti:	-	
Anno corso:	3	
Periodo:	Secondo Semestre	



Testi in italiano

Tipo testo	Testo
Lingua insegnamento	Italiano
Contenuti	Strumenti crittografici: cifrari simmetrici e asimmetrici, funzioni di hash, firma elettronica Sicurezza della rete privata: analisi dei rischi di sicurezza informativa, controllo di accesso, protezione da virus, sistemi firewall, reti private virtuali
Testi di riferimento	Libro di testo: William Stallings: Cryptography and Network Security Prentice Hall. Altri Testi per consultazione: Bruce Schneier: Applied Cryptography, John Wiley and sons, 1994. David Curry: Unix System Security, Addison-Wesley, 1992.
Obiettivi formativi	Il corso si propone di fornire agli studenti gli strumenti crittografici e tecnici utilizzati per garantire la sicurezza di reti e calcolatori. Inoltre, attraverso l'uso di esempi pratici, il corso fornisce agli studenti una comprensione concreta dei maggiori rischi di sicurezza e delle soluzioni disponibili
Prerequisiti	Sistemi Operativi
Propedeutico a	Sicurezza II nella laurea magistrale
Modalità di verifica dell'apprendimento	domande in aula esame scritto
Risultati di apprendimento attesi	conoscenza dei principali argomenti relativi alla sicurezza informatica
Frequenza	non obbligatoria ma fortemente consigliata



Testi in inglese

Tipo testo	Testo
Lingua insegnamento	Italian
Contenuti	Cryptography: symmetric and public key ciphers, hash functions, digital signatures. Network security: risks, access control, malware, firewalls, VPNs.
Testi di riferimento	William Stallings: Cryptography and Network Security Prentice Hall
Obiettivi formativi	The course will provide the main competences required in the field of computer security, including cryptographic algorithms, system and application vulnerabilities, and risk management
Prerequisiti	Operating Systems
Propedeutico a	Security II, in the 'laurea magistrale' programme
Modalità di verifica dell'apprendimento	questions during the lectures, and a written exam
Risultati di apprendimento attesi	knowledge of the main topics regarding computer security
Frequenza	not mandatory, but very useful

Testi del Syllabus

Docente

MEO ROSA

Matricola: **013995**

Anno offerta:

2014/2015

Insegnamento:

MFN0618 - SISTEMI INFORMATIVI (Information Systems)

Corso di studio:

008707 - INFORMATICA (Computer Science)

Anno regolamento:

2012

CFU:

6

Settore:

INF/01

Tipo attività:

D - A scelta dello studente

Partizione studenti:

-

Periodo:

Primo Semestre

Tipo testo	Testo
Lingua insegnamento	italiano
Contenuti	<p>Gli argomenti trattati del corso sono i seguenti:</p> <ol style="list-style-type: none">1. Introduzione ai sistemi informativi aziendali (Modello organizzativo, modello funzionale, modello informatico)2. Sistemi ERP (Le suite ERP; Paradigma ERP; Piattaforme software; Offerta ERP; Trasformazione dell'impresa)3. Integrazione con il cliente: i sistemi CRM (Ruolo dei sistemi CRM nelle aziende; Schema architetturale; Il paradigma; Esempi; Suite di package software; Evoluzione dei CRM)4. Piattaforme di knowledge management (Modello della conoscenza; Sistemi informatici; Progettazione dei sistemi di KM; Modello di successo)5. Piattaforme di Business Intelligence e DSS (Livello delle fonti; Data warehouse; ETL; Progettazione del sistema di warehousing; Livello di elaborazione: reporting e DSS; Motori di analisi e data mining; Suite software per i sistemi direzionali; Sistemi CRM analitici)6. Introduzione al Data Mining e ad alcuni concetti di base (come classificazione e clustering; richiamo di alcuni concetti di statistica di base; piccola sperimentazione di alcuni data-set reali con una suite software open source, come KNIME).
Testi di riferimento	<p>G. Bracchi, C. Francalanci, G. Motta, Sistemi informativi d'impresa, McGraw-Hill, Milano, 2010, ISBN: 978-88-386-6328-4 Libro di consultazione per la parte di Data Warehouse:</p> <p>Ralph Kimball, Margy Ross, The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling, John Wiley & Sons, 2nd Edition, 2002</p>
Obiettivi formativi	<p>Il corso di Sistemi Informativi ha lo scopo di dare una panoramica delle maggiori e più diffuse applicazioni delle basi di dati nel mondo del lavoro e dell'impresa in cui tutti i processi aziendali sono ormai divenuti digitali: i sistemi di pianificazione delle risorse aziendali (Enterprise Resource Planning o ERP), i sistemi integrati di gestione e profilazione del cliente (CRM), i sistemi informativi sanitari, le piattaforme di gestione per la Pubblica Amministrazione (e-government).</p> <p>In particolare si vuole dare una panoramica dei concetti di base che stanno dietro i sistemi di supporto alla decisione (DSS) e alle piattaforme di Business Intelligence: le Data Warehouse e le primitive per l'analisi on-line dei dati (OLAP).</p>
Prerequisiti	<p>Si richiede allo studente di padroneggiare i concetti relativi alle basi di dati, alla programmazione e allo studio degli algoritmi.</p> <p>Si richiede di aver seguito il corso di Basi Dati e Sperimentazioni, i corsi di Programmazione I e Programmazione II, e il corso di Algoritmi e Strutture Dati.</p>
Propedeutico a	Nessuno
Note	Questo corso, con alcune modifiche relative al livello di approfondimento della parte di Data Warehouse verra' seguito anche da studenti dei corsi di Scienze politiche

Tipo testo

Testo

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. Illustrare il concetto di processo aziendale.
2. Spiegare cosa si intende per sistema informativo
3. Descrivere in linguaggio UML il processo aziendale di acquisto di un libro da un catalogo on-line (come Amazon).
4. Illustrare le caratteristiche fondamentali di un sistema ERP.
5. Spiegare il significato di CRM.
6. Spiegare il progetto del modello dei dati per una Data Warehouse in un caso di studio preso da uno dei libri di testo.
7. Discutere alcuni algoritmi di Data Mining.
8. Illustrare uno degli esercizi di analisi dei dati svolti in laboratorio con KNIME.

Risultati di apprendimento attesi

Conoscenza delle principali ambiti di applicazione dei sistemi informativi. Progettazione dei dati e degli algoritmi per i sistemi di supporto alla decisione e l'analisi dei dati.

Conoscenza di una suite di Business Intelligence (analisi dei dati per grossi volumi).

Frequenza

Non obbligatoria ma consigliata



Testi in inglese

Tipo testo	Testo
Lingua insegnamento	italian
Contenuti	<p>The topics of the course are the following:</p> <ol style="list-style-type: none">1. Introduction to Enterprise Information Systems2. ERP systems3. Integration with the client: CRM4. Platforms for knowledge management5. Platforms for Business Intelligence and Decision Support Systems (24 h) (data sources; Data warehouse; ETL process; design of the system for data warehousing; data elaboration: reporting and DSS; data analysis engines and data mining; Software suites; Systems CRM and analytics)6. Introduction to Data Mining with some basic concepts (classification and clustering; recall of some basic concepts of statistics; experiments with an Open Source software suite like KNIME).
Testi di riferimento	<p>G. Bracchi, C. Francalanci, G. Motta, Sistemi informativi d'impresa, McGraw-Hill, Milano, 2010, ISBN: 978-88-386-6328-4</p> <p>On Data Warehouses: Ralph Kimball, Margy Ross, The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling, John Wiley & Sons, 2nd Edition, 2002</p>
Obiettivi formativi	<p>The course of Information Systems has the goal of providing an overview of the main and more wide-spread applications of the databases in the business world and enterprises in which the business processes have become computerized. Examples are the systems for the planning of the enterprise resources (Enterprise Resource Planning o ERP), the integrated systems for the management and profiling of the clients (CRM), the information systems applied to the healthcare, the platforms for the management of the public sector (PA and e-government). In particular, we want to give an overview of the basilar concepts under the decision support systems (DSS) and the Business Intelligence Platforms: the Data Warehouses and the primitives for on-line data analysis (OLAP).</p>
Prerequisiti	<p>Knowledge of basic notions of databases and data representation (such as the Entity-Relationship data model. This will not be required to the students from Political Sciences).</p> <p>Knowledge of basic notions of statistics and algorithms for the part on data analysis.</p>
Propedeutico a	None
Note	<p>This course, with modifications on the Data Warehouse level of study, will be followed also by students coming from the political science Master.</p>
Modalità di verifica dell'apprendimento	<p>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 exercizes seen during the laboratory exercises).</p> <p>As an illustrative example follows here a list of questions that could be done during the oral test:</p>

Tipo testo

Testo

1. Present the concept of business process.
2. Explain the meaning of information system.
3. Present in UML language a business process for the purchase of a book in an on-line catalog (Amazon).
4. Discuss the fundamental characteristics of an ERP system.
5. Explain the meaning of CRM.
6. Explain the data model design for a Data Warehouse tailored for one of the real case studies taken from the course book.
7. Discuss some algorithms of Data Mining.
8. Show the development of some data analysis exercises that were already solved during the laboratory.

Risultati di apprendimento attesi

Knowledge of the main application domains of the information systems.
Design of the data and knowledge of the algorithms at the basis of the decision support systems and of the suites of data analysis.

Frequenza

Non compulsory but suggested.

Testi del Syllabus

Docente	TORASSO PIETRO	Matricola: 010688
Anno offerta:	2014/2015	
Insegnamento:	MFN0607 - SISTEMI INTELLIGENTI (Intelligent Systems)	
Corso di studio:	008707 - INFORMATICA (Computer Science)	
Anno regolamento:	2012	
CFU:	6	
Settore:	INF/01	
Tipo attività:	B - Caratterizzante	
Partizione studenti:	-	
Anno corso:	3	
Periodo:	Secondo Semestre	

Testi in italiano

Tipo testo	Testo
Lingua insegnamento	Italiano
Contenuti	<p>1) INTRODUZIONE ALL'INTELLIGENZA ARTIFICIALE</p> <p>2) RISOLUZIONE AUTOMATICA DI PROBLEMI</p> <ul style="list-style-type: none">- Strategie di ricerca nello spazio degli stati- Strategie di ricerca in presenza di avversario (giochi a due giocatori)- problemi di soddisfacimento di vincoli <p>3) RAPPRESENTAZIONE DELLA CONOSCENZA E RAGIONAMENTO</p> <ul style="list-style-type: none">- Rappresentazione con formalismi logici (calcolo proposizionale e calcolo dei predicati)- Meccanismi di ragionamento (concatenazione in avanti e indietro)- Rappresentazioni ontologiche (il linguaggio OWL 2) <p>3) AGENTI INTELLIGENTI CHE AGISCONO ED IMPARANO</p> <ul style="list-style-type: none">- La nozione di agente intelligente- Architetture di agenti intelligenti- Introduzione all'apprendimento automatico (Alberi di decisioni, reti neurali)
Testi di riferimento	<p>Russell, Norvig: "Artificial Intelligence: a modern approach", Third Edition, Prentice Hall</p> <p>Esiste la traduzione italiana edita da Pearson relativa al primo volume della terza edizione. Rimane valida la traduzione del secondo volume della seconda edizione</p>
Obiettivi formativi	<p>Il corso si propone di fornire una introduzione generale alle problematiche nel settore dell'Intelligenza Artificiale, con particolare attenzione a come sia possibile costruire un sistema dotato di capacità autonome di risoluzione di problemi, di ragionamento e di apprendimento quando abbia a disposizione una rappresentazione simbolica del mondo ed una conoscenza sul mondo stesso (di tipo dichiarativo)</p>
Prerequisiti	<p>Poichè Sistemi Intelligenti è il primo corso che tratta argomenti di Intelligenza Artificiale, le competenze attese in ingresso riguardano competenze nel settore informatico. In particolare:</p> <ul style="list-style-type: none">- conoscenza di algoritmi su alberi e grafi con relative nozioni di complessità- esperienza di programmazione con particolare riferimento a programmazione ad oggetti (organizzazione in classi e sottoclassi, ereditarietà)- nozioni di logica (calcolo proposizionale e calcolo dei predicati del primo ordine)- nozioni di modelli semantici dei dati nelle basi dati
Propedeutico a	<p>Il corso è propedeutico a corsi più avanzati nel settore dell'intelligenza Artificiale offerti nella laurea magistrale in Informatica all'Università di Torino (in particolare Intelligenza Artificiale e laboratorio, Agenti Intelligenti e Sistemi Cognitivi)</p>
Modalità di verifica dell'apprendimento	Esame orale individuale
Frequenza	Non obbligatoria, ma caldamente suggerita



Testi in inglese

Tipo testo	Testo
Lingua insegnamento	Italian
Contenuti	<p>1) INTRODUCTION TO ARTIFICIAL INTELLIGENCE</p> <p>2) AUTOMATED PROBLEM SOLVING</p> <ul style="list-style-type: none">- Search strategies in the state space- Adversarial Search (Two player games)- Constraint satisfaction problems <p>3) KNOWLEDGE REPRESENTATION AND REASONING</p> <ul style="list-style-type: none">- Knowledge representation via logical formalism (propositional calculus and first order predicate calculus)- Reasoning mechanisms (Forward chaining, backward chaining)- Ontological representations (the OWL 2 language) <p>3) INTELLIGENT AGENTS WHICH ACT AND LEARN</p> <ul style="list-style-type: none">- The notion of intelligent agent- Introduction to intelligent agent architecture- Introduction to Machine Learning (Decision trees, Neural networks)
Testi di riferimento	Russell, Norvig: "Artificial Intelligence: a modern approach", Third Edition, Prentice Hall
Obiettivi formativi	<p>The course is aimed at providing a general introduction to the field of Artificial Intelligence. with particular attention to the mechanisms and methodologies enabling the development of an intelligent (artificial systems).</p> <p>The course will emphasize how to design a system with autonomous capabilities of problem solving once a symbolic description of the environment and the problem to be solved.</p> <p>The course will also show how the system is able to perform reasoning once that the world is described in terms of symbolic formalisms and how the system is able to learn its own knowledge when a set of examples is provided.</p>
Prerequisiti	<p>Since Intelligent Systems is an introduction to Artificial Intelligence, the required competences concern the general field of Computer Science. In particular:</p> <ul style="list-style-type: none">- algorithms for graphs and trees (and related complexity)- experience in programming (in particular object oriented)- basic notions on boolean algebra- semantic models for data bases
Propedeutico a	The course provides a solid background for advanced courses on Artificial Intelligence offered in the Master of Computer Science - Università di Torino (in particular, Artificial Intelligence and laboratory, Intelligent Agents, Cognitive Systems)
Modalità di verifica dell'apprendimento	Oral examination
Frequenza	Not compulsory, but strongly suggested

Testi del Syllabus

Docente

GUNETTI DANIELE

Matricola: **013602**

Anno offerta:

2014/2015

Insegnamento:

MFN0601 - SISTEMI OPERATIVI (Operating Systems)

Corso di studio:

008707 - INFORMATICA (Computer Science)

Anno regolamento:

2013

CFU:

12

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

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

Providing a basic knowledge of the architecture and functioning of modern operating systems. Providing an introduction to C language.

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

Prerequisiti

Computer Architecture;
Computer Programming I and II

Propedeutico a

none

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 concurrent programs for the Unix environment.

Frequenza

not mandatory

Testi del Syllabus

Docente	BONO VIVIANA	Matricola: 014039
Anno offerta:	2014/2015	
Insegnamento:	MFN0606 - SVILUPPO DELLE APPLICAZIONI SOFTWARE (Development of Software Applications)	
Corso di studio:	008707 - INFORMATICA (Computer Science)	
Anno regolamento:	2012	
CFU:	9	
Settore:	INF/01	
Tipo attività:	B - Caratterizzante	
Partizione studenti:	-	
Anno corso:	3	
Periodo:	Secondo Semestre	



Testi in italiano

Tipo testo

Testo

Lingua insegnamento

Italiano

Contenuti

Elementi di ingegneria del software (IS):

- cenni alle metodologie:

Waterfall,

V-shaped,

Component-based Software Engineering,

Model driven development,

Incremental,

Spiral,

Issue-based Development,

Rapid Software Development,

Agile,

Extreme Programming,

SCRUM;

- controllo delle versioni;

- testing: unit testing, acceptance test, white e black box testing,

- bug tracking.

Introduzione all'UML:

- use case diagram,

- class diagram,

- object diagram,

- sequence diagram,

- communication diagram,

- state chart,

- activity diagram.

Unified Process (UP).

Questa metodologia verra` descritta in dettaglio e applicata a uno studio di caso nel laboratorio. I passi previsti, in breve, sono:

Pianificazione delle fasi di sviluppo: esse sono suddivise in fasi di ideazione, elaborazione e costruzione, possiamo pensare ad esempio a quattro fasi: una di ideazione, due di elaborazione e una di costruzione. Di tutte le fasi si prevede una durata e i documenti prodotti (tabella delle attivita`).

Prima fase (di ideazione). A partire da una descrizione informale del progetto da sviluppare, si comincia l'analisi dei requisiti con: scelta degli attori e descrizione dei loro obiettivi, individuazione dei casi d'uso (documento prodotto: use case diagram). Si fa poi una suddivisione fra i casi d'uso a seconda della loro priorita` (ad esempio: alta, media, bassa), secondo dei criteri da fissare a priori. Si produce il class diagram di dominio. Si dettagliano i casi d'uso ad alta priorita`, si fanno i sequence di sistema per individuare le operazioni che corrisponderanno a eventi legati all'interfaccia utente e si scrivono i contratti di quelle operazioni

Tipo testo

Testo

che si ritengono piu' complesse. Si producono anche il documento di visione, il documento delle specifiche supplementari, la prima versione del glossario.

Seconda fase (prima di elaborazione). Questa fase concerne principalmente il design dei casi ad alta priorit . Si sceglie la classe Controller (la classe nel sistema che "parla" con l'interfaccia utente). Si dettagliano i corpi delle operazioni dei casi d'uso ad alta priorit  tramite sequence o communication diagram. Si produce un secondo class diagram, in cui si dettagliano campi e metodi e si applicano i design pattern per organizzare il software. Si cominciano a progettare la base di dati e l'interfaccia utente.

In parallelo, si riprende l'analisi. Si rivedono i casi d'uso sviluppati nel passo precedente. Si dettagliano i casi d'uso a priorit  media e il class diagram di dominio. Si fanno i sequence di sistema e i contratti per tali casi d'uso. Si aggiorna il glossario.

Terza fase (seconda di elaborazione). Questa fase concerne principalmente l'implementazione dei casi ad alta priorit  e il design dei casi a media priorit .

Si cominciano quindi a sviluppare gli unit test e l'implementazione dei casi d'uso ad alta priorit .

In parallelo, si producono sequence o communication diagram per i casi a media priorit  e si aggiorna il class diagram di progetto di conseguenza.

Quarta fase (di costruzione). Questa fase concerne principalmente l'implementazione dei casi a media priorit .

In parallelo, e' possibile riprendere l'analisi di ulteriori casi d'uso.

Testi di riferimento

Testo di riferimento per l'introduzione all'IS: Ian Sommerville, Ingegneria del software (Ottava edizione), Addison-Wesley, ISBN 9788871923543.

Testo di riferimento su UP: Craig Larman, Applicare UML e i pattern, Terza edizione (seconda edizione italiana), Pearson.

Esercizi su UML: Bennet, Skelton, Lunn, Introduzione a UML, Collana Schaum's, McGraw-Hill.

Guida minima all'UML: Fowler, UML Distilled, Terza edizione italiana, Pearson-Addison Wesley.

"Bibbia" dei design pattern GoF: Gamma, Helm, Johnson, Vlissides, Prima edizione italiana, Pearson Education Italia-Addison Wesley.

Obiettivi formativi

L'obiettivo principale e` quello di apprendere le idee guida dello sviluppo di applicazioni software modellato con UML e interfacciato con i DBMS, utilizzando la metodologia Agile Unified Process (UP).

Lo studente deve sviluppare un'applicazione significativa individuando con chiarezza la logica applicativa, l'interazione con le basi di dati e le interfacce richieste dai requisiti.

Il lavoro deve essere pianificato secondo i canoni dello sviluppo dei progetti: lavoro di gruppo, definizione degli obiettivi e delle fasi di sviluppo. La presentazione dei risultati dovr  essere compendiativa da una breve relazione.

Il corso ha una forte caratterizzazione sperimentale.

Prerequisiti

Sono consigliati IUM e Tecnologie Web, oppure Programmazione III.

Propedeutico a

NA

Tipo testo**Testo****Note**

-

Modalità di verifica dell'apprendimento

Orale sulla parte metodologica di IS e su UP. Discussione di gruppo e individuale del progetto sviluppato in laboratorio, previa presentazione dei risultati corredati da una breve relazione.

Risultati di apprendimento attesi

Conoscenza pratica della metodologia UP e dell'uso di UML in tale ambito.
Conoscenza di pattern di progettazione e pattern architetture.

Frequenza

Non obbligatoria ma altamente consigliata, in particolare per la parte di laboratorio.

Testi in inglese

Tipo testo

Testo

Lingua insegnamento

Italian

Contenuti

Introduction to Software Engineering (SE):

- some hints to the following methodologies:
Waterfall,
V-shaped,
Component-based Software Engineering,
Model driven development,
Incremental,
Spiral,
Issue-based Development,
Rapid Software Development,
Agile,
Extreme Programming,
SCRUM;

- testing: unit testing, acceptance test, white and black box testing;
- bug tracking;
- version control.

Introduction to UML:

- use case diagram,
- class diagram,
- object diagram,
- sequence diagram,
- communication diagram,
- state chart,
- activity diagram.

Unified Process (UP).

The UP methodology is discussed in detail and applied to a case study in the lab. The main steps are:

Process planning: create a plan for the inception, elaboration, and construction phases of the development process; for example, a plan may involve a total of four phases: inception, two elaboration phases, and construction. For each phase a duration is estimated and the output documents are specified (activity table).

First phase (inception). Starting from an informal description of the system that must be developed, perform an initial requirements analysis: identify actors and their goals, identify use cases (output document: use case diagram). Classify use cases with their priority (e.g., high, medium, low), based on criteria defined a priori. Create the domain class diagram. Detail the high-priority use cases, and create system sequence diagrams in order to identify the operations corresponding to UI events; write contracts for the most complex operations. Additional output documents

Tipo testo

Testo

are the vision document, additional specs, and glossary.

Second phase (first of elaboration). Design of high-priority cases. Choice of the Controller class (the class of the system that "communicates" with the UI). The bodies of the operations in high-priority use cases are detailed through sequence or communication diagrams. A second class diagram, which details fields and methods of the classes, is created; design pattern are applied to organize the classes. The UI and DB design starts.

At the same time, the analysis continues. The use cases that have already been detailed are reviewed. Medium-priority use cases are detailed, the domain class diagram is updated. Sequence diagrams and contracts for such use cases are created. The glossary is updated.

Third phase (second of elaboration). Design of medium-priority cases and implementation of high-priority cases.

The implementation and unit test of the high-priority use cases starts. At the same time, sequence or communication diagrams for medium-priority cases are produced, and the class diagram is updated accordingly.

Fourth phase (construction). Implementation of medium-priority cases. At the same time, further use cases may be analyzed.

Testi di riferimento

Reference for the introduction to SE: Ian Sommerville, Ingegneria del software (Ottava edizione), Addison-Wesley, ISBN 9788871923543.

Reference on UP: Craig Larman, Applicare UML e i pattern, Terza edizione (seconda edizione italiana), Pearson.

Exercises on UML: Bennet, Skelton, Lunn, Introduzione a UML, Collana Schaum's, McGraw-Hill.

Minimal guide to UML: Fowler, UML Distilled, Terza edizione italiana, Pearson-Addison Wesley.

GoF design pattern "bible": Gamma, Helm, Johnson, Vlissides, Prima edizione italiana, Pearson Education Italia-Addison Wesley.

Obiettivi formativi

The main goal is learning the guideline of the development of software applications modelled with the aid of UML and interfaced with DBSM, by means of the Agile methodology Unified Process (UP).

The student must develop a significant application, by singling out the applicative logic, the interactions with the data base, and the required user interfaces.

The work must be planned according to the guidelines of project development: group work, definition of objectives and of the development phases. The presentation of the results must be accompanied by a synthetic report.

This course has a strong practical connotation.

Prerequisiti

IUM and Tecnologie Web, or Programmazione III are recommended.

Propedeutico a

NA

Note

-

Tipo testo**Testo****Modalità di verifica dell'apprendimento**

Oral test on the theoretical part of SE and on UP. Group and individual presentation of the project, together with a synthetic report on the results.

Risultati di apprendimento attesi

Practical knowledge of the UP methodology and of UML usage in this setting. Knowledge of design and architectural patterns.

Frequenza

It is not mandatory, but highly recommended for the practical part.

Testi del Syllabus

Docente	RUFFO GIANCARLO FRANCESCO Matricola: 014114
Anno offerta:	2014/2015
Insegnamento:	MFN0634 - TECNOLOGIE WEB (Web Technologies)
Corso di studio:	008707 - INFORMATICA (Computer Science)
Anno regolamento:	2012
CFU:	6
Settore:	INF/01
Tipo attività:	B - Caratterizzante
Partizione studenti:	-
Anno corso:	3
Periodo:	Primo Semestre

 **Testi in italiano**

Tipo testo	Testo
Lingua insegnamento	Italiano
Contenuti	Architetture client-server Paradigma Model-View-Controller Protocollo Http Html e css Javascript Programmazione web lato server: cookies, sessioni PHP Interazione con DB MySql Aspetti di sicurezza: SQL injection, Cross Site Scripting (cenni) Programmazione client-side avanzata: Ajax, librerie javascript di visualizzazione Cenni di Sviluppo per Smartphone (iOS, Android)
Testi di riferimento	V. Della Mea, L. Di Gaspero, I. Scagnetto, Programmazione Web lato server, Apogeo
Obiettivi formativi	<p>Il corso verte sull'architettura di applicazioni client server, in particolare su quelle web basate su HTTP. Questo corso affronta inoltre la progettazione e lo sviluppo di servizi accessibili in internet, integrando aspetti teorici e sperimentazione in laboratorio.</p> <p>Ci si attende che al termine del corso lo studente sappia: analizzare, progettare e realizzare applicazioni distribuite in architettura web client/server, sia a livello client che a livello server.</p>
Prerequisiti	Programmazione I e II, Architetture, Sistemi Operativi, Reti, Data Base.
Propedeutico a	laurea
Modalità di verifica dell'apprendimento	<p>Il controllo sull'apprendimento e' basato sia sull'interazione con gli studenti nel corso delle lezioni e sia assegnando agli studenti un progetto da realizzare, suddiviso in piu' fasi, con controlli periodici del livello di avanzamento.</p> <p>L'esame finale e' costituito da un progetto di laboratorio (i dettagli saranno disponibili sulla pagina web del corso) e dalla discussione dello stesso in forma orale.</p>
Risultati di apprendimento attesi	Al termine del corso lo studente sapra' analizzare, progettare e realizzare applicazioni distribuite in architettura web client/server, sia a livello client che a livello server; avra' le basi tecniche per intraprendere la strada della commercializzazione di applicazioni per il Web.



Testi in inglese

Tipo testo

Testo

Lingua insegnamento

Italian

Contenuti

Client-server architectures
Model-View-Controller paradigm
Http protocol
Html and css
Javascript
server side programming: cookies, sessions
PHP
interaction with DB MySql
Web sicurezza: SQL injection, Cross Site Scripting
Advanced client-side programming: Ajax, visualization with javascript libraries
Survey on development of application for smartphones (iOS, Android)

Testi di riferimento

V. Della Mea, L. Di Gaspero, I. Scagnetto, Programmazione Web lato server, Apogeo
