REVISIONE 1.1 – 15/10/2019: syllabus insegnamenti in appendice; docenti tabella seconda parte pag. 24

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

Corso di laurea in Informatica

Dipartimento di Informatica

Università degli Studi di Torino

Via Pessinetto 12 - Torino

Anno Accademico 2019/2020

Prefazione

Gentile lettrice, gentile lettore,
questa è la guida del corso di Laurea 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 in Informatica (DM 270, classe di laurea L31), questa descrizione costituisce il "Manifesto degli studi - Piano dell'Offerta Formativa sui tre anni per la coorte 2019", mentre la seconda parte descrive tutti gli insegnamenti offerti per il 2019/2020 (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 e le studentesse che si immatricolano quest'anno (coorte 2019), che trovano nel Manifesto la descrizione del loro percorso triennale, mentre la seconda parte è di interesse anche per gli studenti e le studentesse delle due coorti precedenti (2017 e 2018) che devono seguire gli insegnamenti del terzo e secondo anno, rispettivamente. Ricordiamo che per gli studenti e le studentesse immatricolati negli anni

precedenti vale il manifesto degli studi della loro coorte, già pubblicato negli anni precedenti.
Le attività del Corso di Laurea (CdL) di Informatica dell'Università di Torino sono coordinate dal Consiglio di Corso di Studi (CCS), composto dai docenti che insegnano nel CdL e dai rappresentanti degli studenti, periodicamente eletti. Le attività del CCS sono normate dal ``Regolamento del Corso di Laurea in Informatica''.

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

1. Tutto quanto è scritto in questa guida (e nei successivi, eventuali aggiornamenti alla guida stessa che verranno pubblicati sul sito del Corso di Laurea) è vincolante per gli studenti e le studentesse della coorte 2019 e per il corso di laurea e il suo corpo docente. La guida, unitamente ai regolamenti e alle scadenze di Ateneo per tasse, piano carriera ed esame di laurea, costituisce la base del patto "studenti-Università": conoscerla e averne compreso i contenuti è un obbligo dello studente/della studentessa.
2. Il Corso non è a numero programmato, ma tutti gli studenti e le studentesse devono sostenere e superare

il TARM (test di accertamento dei requisiti minimi) per poter sostenere gli esami dei singoli insegnamenti. **Il TARM va effettuato prima di iscriversi**, perché tale test è utile anche come sistema di autovalutazione delle proprie capacità rispetto al corso di studio. Leggete con cura la parte della guida che spiega motivazioni, modalità e sanzioni (paragrafo su "Requisiti di ammissione").

3. 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". Tutti i servizi agli studenti e alle studentesse (segreteria studenti, pagamento tasse e altro) sono supportati da un servizio di help-desk a cui gli studenti/le studentesse sono invitati a rivolgersi prima di andare allo sportello o di telefonare. L'help-desk fornisce una risposta scritta e quindi può essere uno strumento utile per risolvere eventuali incomprensioni.
4. Tutte le informazioni non riportate in questa guida (come i programmi dettagliati degli insegnamenti, gli orari e la localizzazione delle aule) sono reperibili dal

sito Internet del Corso di Laurea, all'indirizzo http://laurea.educ.di.unito.it/

5. Il corso di Laurea in Informatica è uno dei corsi di Laurea della Scuola di Scienze della Natura dell'Università di Torino
6. La durata della Laurea è di tre anni accademici. I primi due anni sono uguali per tutti gli studenti e le studentesse: gli insegnamenti dei primi due anni sono suddivisi fra corso A e corso B e gli insegnamenti di laboratorio sono poi ulteriormente suddivisi, per permettere un adeguato rapporto docente/studenti. Il terzo anno è invece articolato in tre diverse aree tematiche di approfondimento, al fine di permettere allo studente/alla studentessa una maggiore personalizzazione degli studi.
7. La Laurea in Informatica dà accesso alla Laurea Magistrale in Informatica, della durata di due anni accademici, laurea che, al momento, è anch'essa strutturata in tre percorsi.
8. La Laurea 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.

9. Il termine "Corso di Laurea" è usato in questa Guida per riferirsi al Corso di Laurea in Informatica (triennale) della classe L31 (Scienze e Tecnologie Informatiche) attivato dall'Università di Torino e con sede presso il Dipartimento di Informatica.
10. L'Università di Torino adotta una procedura telematica per l'iscrizione agli esami e la registrazione dei voti conseguiti (con conseguente abolizione del "libretto degli esami" in forma cartacea). Per ogni esame, nonché per la registrazione del superamento dello stage, è obbligatoria l'iscrizione.
11. La valutazione degli insegnamenti avviene con procedura telematica quando si aprono le cosiddette "finestre di valutazione", tipicamente fra la fine del semestre e la fine degli appelli di esame. La valutazione di un insegnamento da parte dello studente/della studentessa è condizione necessaria per l'iscrizione all'esame dell'insegnamento stesso. Gli studenti e le studentesse possono poi esprimere una loro valutazione sulle modalità di esame e sulla congruenza fra esame e insegnamento nelle valutazioni dell'anno successivo.

12. La documentazione completa del Corso di Studi, così come caricata annualmente sul sito del Ministero dell'Istruzione, l'Università e la Ricerca (MIUR), la potete trovare alla voce Scheda Unica Ateneo, Laurea L31, Università di Torino, sul sito MIUR www.universitaly.it

In ultimo, una raccomandazione e un consiglio per gli immatricolandi: è molto importante che gli studenti e le studentesse 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, sostenendo 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 Laurea più semplice, più interessante e più coinvolgente. Gli studenti e le studentesse 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 Laurea che si terrà il giorno ***20 settembre 2019,*** alle ore 14.00 in aula A e B, Dipartimento di Informatica, con accesso da via Pessinetto 12 - Torino. La presentazione è aperta anche agli studenti e alle studentesse 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. Spero che l'Università non sia per voi il luogo degli esami, ma il luogo dell'apprendimento e della crescita personale e sociale. Concentrarsi sull'apprendimento e sulle competenze, e sulle multiformi possibilità che l'Università vi offrirà per acquisirle, sarà il modo più facile e divertente di portare a termine i vostri Studi con successo.

Buona lettura!

Maria Luisa Sapino, PhD,
Professoressa Ordinaria di
Informatica,
Presidente del Corso di Laurea e
Laurea Magistrale in Informatica

Tel. 011 6706745

e-mail

presccs@educ.di.unito.it

INDICE

PREFAZIONE **2**

PRIMA PARTE: MANIFESTO DEGLI STUDI **14**

Obiettivi e sbocchi professionali **14**

Obiettivi del Corso di Laurea 14

Sbocchi professionali 15

Certificazione di Qualità 16

Organizzazione della didattica **18**

Introduzione 18

Requisiti di ammissione 20

Iscrizione come seconda laurea, passaggio o trasferimento. 23

Pre-corsi (corsi di riallineamento) 24

Iscrizione a tempo pieno e a tempo parziale 25

Piano Carriera e responsabili 26

Date dei periodi di lezione 27

Esami 29

Il processo della qualità del Corso di Laurea 31

Riconoscimento Certificazioni di lingua inglese 32

Il tutorato 33

Date e scadenze 34

La Laurea in Informatica 34
Struttura del Corso di Laurea 35
Biennio propedeutico comune 40
Area tematica di approfondimento Informazione e Conoscenza 42
Area tematica di approfondimento Linguaggi e Sistemi . 45
Area tematica di approfondimento Reti e Sistemi Informatici 48
Struttura dell'offerta formativa rispetto al RAD della classe 51

Supporto agli studi 55
Erasmus 55
Biblioteca 57
Supporto on-line agli insegnamenti 58
Aule e laboratori 58
Zone studio 60
Supporto per gli studenti Disabili 61
Supporto per gli studenti con DSA (Disturbi Specifici di Apprendimento) 64

Indirizzi utili 67

SECONDA PARTE: PROGRAMMI E DOCENTI DEGLI INSEGNAMENTI PER L'A.A. 19/20 70

Insegnamenti attivi nel 19/20. 70
Codici per convalide di insegnamenti e competenze. ... 81
Programmi e altre informazioni per gli insegnamenti attivi nel 19/20 (syllabus degli insegnamenti) 83

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 Laurea fornisce una buona preparazione metodologica di base, coniugata con la comprensione delle più recenti tecnologie in diversi ambiti, dall'impresa al servizio pubblico e alle applicazioni individuali. Obiettivo del corso di laurea 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.

I laureati in Informatica saranno in grado di utilizzare le conoscenze e le competenze acquisite nella progettazione,

sviluppo e gestione di sistemi informatici; possiederanno inoltre le capacità necessarie per affrontare e analizzare problemi in vari contesti applicativi e per sviluppare sistemi informatici per la loro soluzione.

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

Sbocchi professionali

I laureati e le laureate 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 laurea consentono

di avviare attività professionali autonome, prospettiva che oggi trova sempre più spazio nel mondo del lavoro.

Il corso prepara alle seguenti **professioni ISTAT**: Analisti e progettisti di software, Analisti di sistema, Analisti e progettisti di applicazioni web, Specialisti in reti e comunicazioni informatiche, Analisti e progettisti di basi dati, Amministratori di sistemi, Tecnici programmatori, Tecnici esperti in applicazioni, Tecnici web, Tecnici gestori di basi di dati, Tecnici gestori di reti e di sistemi telematici.

Il D.P.R. 328/2001 ha istituito nell'Albo professionale degli Ingegneri le Sezioni A e B, in relazione al diverso grado di capacità e competenza acquisita mediante il percorso universitario (magistrale e triennale rispettivamente). Ciascuna sezione è ripartita nei seguenti Settori: Civile e Ambientale, Industriale e dell'Informazione. La Laurea in Informatica è una delle lauree previste 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, dal 2004 e in intesa con AICA (Associazione Italiana per il Calcolo Automatico), un certificato che attesta la qualità dei Corsi di Laurea e Laurea Magistrale in Informatica italiani. Il nostro Corso di Laurea ha già ricevuto tale certificato nazionale di qualità negli anni scorsi e in particolare nel 2018, l'ultimo al momento attivato. Il Corso di Laurea sta lavorando alla redazione dei documenti necessari alla richiesta del bollino 2019, che verrà effettuata non appena la procedura sarà disponibile.

Informazioni più dettagliate sui criteri e gli obiettivi di questa certificazione si possono ottenere direttamente sul sito GRIN.

Il nostro corso di laurea ha partecipato alle attività per l'accreditamento ministeriale dell'Università di Torino, primo grande Ateneo italiano a candidarsi per l'accreditamento. In particolare il nostro corso di laurea è stato uno dei 7 corsi selezionati dai valutatori per le visite di accreditamento che si sono tenute nel novembre 2015

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 sé 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 o europei, non ha una cadenza, una data di inizio e un ambito tematico ricorrente, e quindi le informazioni non sono riportate in questa guida. 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* fornisce conoscenze di base ad ampio spettro, accanto a elementi di formazione professionalizzante, al fine di consentire il proseguimento negli studi magistrali, ma anche 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 o semplicemente "credito" nel prosieguo). Le norme di legge prevedono che ogni CFU equivalga a 25 ore di lavoro per lo studente/la studentessa. 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 Laurea in Informatica il Regolamento Didattico stabilisce che ogni CFU sia di norma equivalente a 8 ore di lezione/10 ore di esercitazione in aula o in laboratorio + 17/15 ore di studio individuale.

Ad esempio, un insegnamento di 6 crediti quale, Calcolo Matriciale e Ricerca Operativa (primo semestre del primo

anno) corrisponderà a 48 ore di lezioni ed esercitazioni, e si assume che allo studente e alla studentessa (che abbiano una buona conoscenza delle tematiche propedeutiche per il corso, indicati nel seguito come requisiti di base) siano richieste ulteriori 102 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 a studenti e studentesse che abbiano 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 e le studentesse in possesso di Diploma di Scuola Media Superiore o titolo equivalente. Per frequentare il Corso di Laurea in Informatica non si richiedono competenze informatiche precedenti, 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 tutti gli studenti e le studentesse che desiderano iscriversi a Informatica.

L'ateneo di Torino sulla base di lunga esperienza di proposte di TARM diversificate e basate per lo più su tematiche disciplinari, alla luce dei risultati ottenuti dagli studenti e dalle studentesse in termini di progressione della carriere, e in modo particolare di acquisizione di 40 CFU nel primo anno, monitorate attentamente negli ultimi 3 anni accademici, con un'analisi supportata anche da un gruppo di esperti, ha concluso che la capacità di avanzamento nel percorso universitario sia fondato sulla capacità di comprensione di testi universitari e su basi adeguate di matematica e lingua italiana.

A seguito di questa analisi i requisiti di accesso saranno accertati tramite una prova unica di Ateneo idonea a valutare questi aspetti e chi ancora non possiede tali abilità dovrà rafforzare le proprie soft skills (https://www.almalaurea.it/info/aiuto/lau/manuale/soft-

skill) attraverso un percorso unico articolato mediante formazione erogata in modalità telematica che prevede una graduale acquisizione delle soft-skill utili alla frequenza e allo studio con profitto dei corsi universitari (https://www.passport.unito.it/).

L'iscrizione al Corso di Laurea, pertanto, sarà subordinata allo svolgimento del Test di Accertamento dei Requisiti Minimi (TARM) che consisterà in una prova a risposta multipla con un numero complessivo di 55 domande, ciascuna con più risposte possibili di cui 1 esatta, da svolgersi entro il tempo massimo di 120 minuti.

In particolare il TARM ha una struttura articolata in 6 sezioni per un totale di 55 domande e una soglia a 30:

20 domande di comprensione del testo
10 domande di matematica
10 domande di lingua italiana
5 domande di cultura generale umanistica
5 domande di cultura generale scientifica
5 domande di cultura generale giuridica, economica e sociale

A seguito di esito insufficiente del test verranno assegnati degli OFA (Obblighi Formativi Aggiuntivi) che consistono nel percorso Passport.U. Gli OFA sono considerati assolti con il completamento del percorso unico di rafforzamento delle soft skills, da completarsi, a iscrizione avvenuta, per poter procedere alla compilazione del piano carriera

Iscrizione come seconda laurea, passaggio o trasferimento.

Gli studenti e le studentesse che intendono iscriversi come seconda laurea, passaggio da altro corsi di laurea 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, alla seguente commissione:

Commissione	E-mail
Commissione Passaggi, Trasferimenti, II lauree e piani di studio	commpassccs@educ.di.unito.it

Per maggiori informazioni consultare il sito di Ateneo nella sezione "passaggi e trasferimenti".

Ricordiamo che in generale anche per questi studenti e studentesse è obbligatorio sostenere il TARM.
Sono però previsti casi di esonero, per conoscere i quali vi invitiamo a consultare la pagina https://www.unito.it/didattica/immatricolazioni-e-iscrizioni/corsi-ad-accesso-libero .

Pre-corsi (corsi di riallineamento)

I pre-corsi sono uno strumento che il corso di laurea mette a disposizione degli studenti e delle studentesse che intendano ripassare e re-inquadrare le competenze di matematica e fisica da loro acquisite nelle scuole superiori, in aggiunta a quanto fatto per il superamento del TARM. I pre-corsi sono associati ai normali insegnamenti: per Analisi Matematica si svolgeranno nelle prime settimane di lezione del secondo semestre, all'interno dell'orario dell'insegnamento di Analisi Matematica (primo anno, secondo semestre). Per Fisica il pre-corso è invece previsto al secondo anno, durante la prima settimana di lezione dell'insegnamento di Fisica.

Iscrizione a tempo pieno e a tempo parziale

La riforma 270 ha introdotto una distinzione importante tra gli studenti e le studentesse universitari: quelli iscritti a tempo pieno e quelli iscritti a tempo parziale. Gli studenti e le studentesse "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 Laurea, si può osservare che in media ciascuno studente e ciascuna studentessa deve acquisire 60 crediti all'anno, che complessivamente corrispondono a 1500 (=60*25) ore di lavoro.

Se però uno studente o una studentessa 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/la studentessa non si trova costretto/costretta a seguire un ritmo di studio eccessivo, che lo/la 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 gli studenti e le studentesse iscritti a tempo pieno potranno registrare un massimo di 80 CFU all'anno, mentre gli studenti e le studentesse iscritti a tempo parziale potranno registrare un massimo di 36 CFU all'anno.

Piano Carriera e responsabili

Gli studenti e le studentesse sono tenuti a presentare il piano carriera (ex carico didattico), cioè l'elenco di tutti gli insegnamenti del proprio piano di studi, seguendo le istruzioni presenti sulla pagina pubblica e personale del sito dell'Ateneo (www.unito.it): l'applicativo per il piano carriera viene di norma reso disponibile fra ottobre e gennaio. Il piano carriera si compila secondo le modalità e le scadenze descritte sul sito di Ateneo, e in **accordo con l'offerta formativa della propria coorte e le regole dei piani carriera ad essa associate**, che specificano gli insegnamenti obbligatori e a scelta, e la loro collocazione nei tre anni.

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

Per ulteriori informazioni e per modifiche dei piani carriera al di fuori delle scelte già previste dall'offerta formativa, gli studenti e le studentesse devono far riferimento alla *Commissione passaggi, trasferimenti e piani di studio* del Corso di Laurea di Informatica (commpassccs@educ.di.unito.it). Gli studenti e le studentesse devono contattare la commissione almeno due settimane prima della scadenza per la presentazione del piano carriera. Per ulteriori informazioni sui percorsi gli studenti e le studentesse possono rivolgersi ai relativi docenti responsabili:

- area tematica di approfondimento "Informazione e Conoscenza": Liliana Ardissono (liliana@di.unito.it)
- area tematica di approfondimento "Linguaggi e Sistemi": Viviana Bono (bono@di.unito.it)
- area tematica di approfondimento "Reti e Sistemi Informatici": Giancarlo Ruffo (ruffo@di.unito.it)

Date dei periodi di lezione

Per il Corso di Laurea in Informatica l'Anno Accademico è

suddiviso in due periodi didattici (semestri).

Per l'Anno Accademico 2019/2020 le date di inizio e di fine del periodo di lezioni/esami sono le seguenti:

Primo semestre I anno: lunedì 23 settembre 2019 – venerdì 20 dicembre 2019 (eventuali recuperi 7-17 gennaio 2020)

Primo semestre II-III anno: martedì 24 settembre 2019 – venerdì 6 dicembre 2019
+ martedì 7 gennaio 2020 – venerdì 17 gennaio 2020 (eventuali recuperi 20-22 gennaio 2020)

Sessione straordinaria per appelli del II semestre 2018-19 (studenti II-III anno e fuori corso) 9-20 dicembre 2019

Sessione invernale restanti appelli per tutti: dal 23 gennaio 2020 al 25 febbraio 2020

Interruzione natalizia per tutti da sabato 21 dicembre 2019 a lunedì 6 gennaio 2020, estremi compresi

Secondo semestre (I, II e III anno): mercoledì 26 febbraio 2020 – venerdì 5 giugno 2020

Interruzione pasquale per tutti: da giovedì 9 aprile 2020 a martedì 14 aprile 2020, estremi compresi

Sessione estiva di esami: da lunedì 8 giugno 2020 a venerdì 31 luglio 2020

Sessione autunnale di esami: da lunedì 31 agosto 2020 – inizio aa 2020-21

Gli orari e le aule degli insegnamenti sono disponibili all'indirizzo http://di.unito.it/orarilezioni. La coorte degli studenti e delle studentesse viene divisa in due per le lezioni frontali (corso A, studenti e studentesse con iniziale del cognome A-K e corso B, studenti e studentesse con iniziale del cognome L-Z) e in quattro per le lezioni di laboratorio (corso A - turno T1, corso A - turno T2, corso B - turno T1, corso B, turno T2). La regola di suddivisione dei laboratori è descritta sulle pagine web dei singoli insegnamenti (di.unito.it/insegnamenti).

Esami

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 *ciascuno studente e ciascuna studentessa deve obbligatoriamente iscriversi* allo stesso dal proprio sito della bacheca degli esami, disponibile sulla pagina personale dello studente e della studentessa sul sito di Ateneo. Le modalità di esame variano da insegnamento a insegnamento, ma rimangono invariate per tutti e 5 gli appelli dell'anno di uno stesso insegnamento. È possibile iscriversi all'esame di un insegnamento solo se

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

Gli studenti e le studentesse della coorte 2019, quando saranno iscritti al secondo o al terzo anno, potranno sostenere esami relativi a un anno superiore al primo solo dopo aver superato esami del primo anno per un numero di crediti complessivo pari ad almeno 21.

Sono previste tre sessioni di laurea l'anno, normalmente nei mesi di luglio, ottobre (con prolungamento a dicembre) - ultima sessione valida per l'iscrizione alla magistrale, e Marzo/Aprile. Le date per le sessioni di laurea della coorte 2018 saranno rese note entro l'inizio del terzo anno accademico della coorte. Il regolamento di Ateneo prevede che ciascuno studente e ciascuna studentessa debba terminare gli esami al momento della domanda di laurea. Per i soli studenti e studentesse in corso, e limitatamente alle sessioni di luglio e ottobre, è previsto di poter terminare gli esami anche dopo la scadenza della domanda,

sino a 10 giorni prima della data dell'esame di laurea. Queste scadenze hanno un particolare impatto sugli studenti e sulle studentesse fuori corso (cioè iscritti da più di 3 anni) per i quali non sarà garantito che possano usufruire degli appelli di esame di giugno/luglio per potersi laureare a luglio e di settembre per potersi laureare ad ottobre.

Il processo della qualità del Corso di Laurea

La valutazione della didattica è obbligatoria per tutti gli studenti e tutte le studentesse e per tutti i/le docenti. Per poter sostenere l'esame di un insegnamento lo studente e la studentessa devono aver *obbligatoriamente valutato* l'insegnamento stesso. Oggetto della valutazione saranno l'interesse per l'insegnamento, la chiarezza espositiva del docente, la regolarità nello svolgimento delle lezioni, le strutture (aule o laboratori) in cui il corso si è tenuto. Tale procedura, supportata dall'applicativo Edumeter, è obbligatoria anche per gli studenti e le studentesse non frequentanti (che avranno un diverso insieme di domande) e per i/le docenti (che potranno esprimere un parere sulle strutture e sulla preparazione e interesse degli studenti e delle studentesse). Inoltre, una volta l'anno, gli studenti e le studentesse devono valutare gli insegnamenti che hanno

sostenuto nell'anno precedente rispetto alle modalità di esame e alla corrispondenza fra contenuti dell'esame e programma dell'insegnamento.
Le valutazioni sono raccolte in forma anonima e costituiscono la base per le attività di monitoraggio della didattica erogata dal nostro Corso di Laurea. In particolare ogni anno, in autunno, il corso di studio effettua un'attività di monitoraggio annuale delle attività di insegnamento: si analizzano gli eventuali punti di forza e le criticità, si identificano contromisure e le si pianificano nel tempo, inoltre si verifica se le contromisure precedentemente identificate siano state correttamente implementate o siano in fase di svolgimento. Alle attività del monitoraggio partecipano anche i rappresentanti degli studenti o un loro delegato. Eventuali problemi con l'applicativo web di valutazione possono essere risolti contattando il servizio di help-desk: www.unito.it/servizi_federati

Riconoscimento Certificazioni di lingua inglese

Gli studenti e le studentesse 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 on-line sul proprio spazio MyUniTO oppure alla Segreteria Studenti del Polo delle Scienze della Natura, in via S. Croce 6, secondo le scadenze definite dalla Segreteria, di norma a metà giugno di ogni anno.
A seguito di apposita delibera, gli esami riconosciuti saranno direttamente caricati sulla carriera degli studenti e delle studentesse, che non dovranno 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 e assistere gli studenti e le studentesse lungo tutto il percorso degli studi, a renderli attivamente partecipi del processo formativo, a rimuovere gli ostacoli a 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 Laurea in Informatica ha attivato il 'servizio' di tutorato individuale. Si tratta di un'attività che si propone di aiutare gli studenti

e le studentesse a procedere con regolarità negli studi. A tale scopo, a ogni studente e studentessa viene assegnato un docente di riferimento (*tutor*). Gli studenti e le studentesse partecipano agli incontri di tutorato nei tempi e nei modi stabiliti dalla commissione tutorato. Per il Tutorato sono previsti due incontri, uno a fine della prima sessione di esami e uno all'inizio dell'anno successivo. Il secondo incontro è particolarmente utile per pianificare il prosieguo degli studi qualora gli studenti o le studentesse abbiano incontrato delle difficoltà e non abbiano acquisito tutti i crediti formativi previsti per il primo anno. Il calendario degli incontri viene comunicato con mail a tutti gli studenti e a tutte le studentesse.

Date e scadenze

Per tutte le date, le scadenze e le modalità di immatricolazione e passaggi si faccia riferimento all'apposita sezione del *sito web dell'Ateneo*.

La Laurea in Informatica

Nel seguito viene descritta la struttura del corso di laurea, comune alle tre aree tematiche di approfondimento, e come tale struttura sia declinata negli insegnamenti attivati per l'A.A. 2019/2020. Vengono come prima cosa delineati alcuni aspetti comuni alle tre aree, 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 percorsi, con l'elenco degli insegnamenti associati e la loro distinzione in caratterizzanti, affini e liberi, e la loro natura obbligatoria o meno.

Struttura del Corso di Laurea

Il Corso di Laurea in Informatica si articola in tre diverse aree tematiche di approfondimento con i primi due anni in comune, dedicati allo studio di materie propedeutiche per gli insegnamenti successivi. Nel primo e secondo anno sono presenti insegnamenti di matematica, insegnamenti di introduzione alla struttura dei calcolatori, alla programmazione, ai sistemi operativi, agli algoritmi e alle basi di dati, oltre a complementi di economia, diritto e fisica.

Il terzo anno gli studenti e le studentesse dovranno scegliere tra le tre aree tematiche di approfondimento: ***Informazione e Conoscenza*** (*IC*), ***Linguaggi e Sistemi*** (*LS*), e ***Sistemi e Reti*** (*SR*). Tutte le aree tematiche di approfondimento affiancano alla didattica svolta in aula delle ore di didattica e di esercitazioni svolte 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 con la supervisione di un tutor aziendale e di un tutor accademico, che offre agli studenti e alle studentesse 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/la studentessa) sia di norma equivalente a 8 ore di lezione/esercitazione più 17 ore di studio individuale. Per facilitare l'inserimento nel mondo universitario, si propone un primo anno più leggero, di 57 CFU, seguito da due anni di 60 e 63 CFU. Ricordiamo che, nella definizione del piano carriera 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 degli insegnamenti, che permettono di avere un quadro complessivo dei diversi anni. Maggiori dettagli (semestre di svolgimento, docenti) si possono trovare nelle parte II di questa guida, mentre le schede di tutti gli insegnamenti sono presenti nell'appendice (syllabus), oltre che, naturalmente, nel sito del Corso di Laurea (http://www.informatica.unito.it), nella parte relativa alla "Offerta formativa".

La laurea prevede 180 CFU. Nelle tabelle che seguono, che descrivono le unità didattiche, la somma totale dei CFU (primo e secondo anno in comune più terzo anno, specifico per ciascuno dei *percorsi*) è pari a 156. I restanti 24 CFU sono così suddivisi: 9 CFU per lo stage e 3 per la prova finale (si veda il paragrafo successivo) e 12 CFU liberi a scelta dello studente/della studentessa (6 dei quali possono essere dedicati a un'estensione dello stage).

Stage e prova finale

Per completare gli studi, lo studente/la studentessa dovrà effettuare un periodo di stage, la cui durata è

proporzionale al numero di crediti, 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). Lo stage può essere svolto anche all'interno dell'Università, su specifici progetti all'uopo identificati. Al termine dello stage, lo studente/la studentessa 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 la prova finale (esame di laurea). Lo stage terminato con successo porta al conseguimento degli associati CFU, da registrarsi sul libretto elettronico come ogni altro esame, 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 3 CFU.

La stage standard è da 9 cfu (corrispondenti a circa 225 ore in azienda), ma lo studente/la studentessa ha facoltà di aumentare l'impegno dedicato allo stage scegliendo di dedicare 6 dei 12 CFU "liberi – a scelta dello studente'' allo stage, che diventa così da 15 CFU (corrispondenti a circa 375 ore in azienda).

Per poter registrare i CFU corrispondenti allo stage, sia interno che esterno all'Università, la procedura richiede allo studente/alla studentessa di iscriversi all'appello "stage" sulla piattaforma Esse3 relativa agli appelli d'esame e di inviare una mail al proprio relatore per comunicare l'avvenuta iscrizione. Nel caso di stage da 15 CFU ci si dovrà iscrivere anche all'appello "prolungamento stage". I 3 CFU di Prova Finale si acquisiscono invece il giorno della laurea, senza necessità di preventiva registrazione.

CFU a scelta libera dello studente

Lo studente/la studentessa 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 percorso troverete indicate le scelte per i CFU a scelta libera dello studente/della studentessa che sono automaticamente approvate, senza necessità di convalida. Lo studente/la studentessa può usare i CFU liberi anche allo scopo di acquisire crediti in settori non informatici, per esempio per soddisfare requisiti minimi di accesso ad altre lauree magistrali.

Biennio propedeutico comune

Nelle tabella il numero di CFU indicato come n+x indica che il corso si compone di n CFU di lezioni e x CFU di lezioni/attività di laboratorio

I anno

Codice	**Insegnamento**	**CFU**
	Uno a scelta tra i seguenti due:	
mfn0578	Matematica discreta e logica	12
mfn0143	Matematica discreta e logica ON LINE (Chi avesse seguito in precedenza il modulo di "Elementi di Logica" in modalità on-line come progetto Start@Unito deve selezionare l'insegnamento MATEMATICA DISCRETA E LOGICA (ON-LINE) (MAT0143); chi invece non ha mai seguito alcun modulo di tale Progetto deve selezionare MFN0578 MATEMATICA DISCRETA E LOGICA. L'appello del corso Start@Unito "Elementi di Logica" e della parte di Logica del corso "Matematica Discreta e Logica" è unico e condiviso. Gli studenti sono invitati a iscriversi al corso registrato sul proprio	12

	piano carriera. I crediti acquisiti col superamento della prova d'esame saranno attribuiti conformemente.)	
Codice	**Insegnamento**	**CFU**
mfn0570	Analisi Matematica	9
mfn0582	Programmazione I	6 + 3
mfn0585	Programmazione II	6 + 3
mfn0586	Architettura degli elaboratori	6 + 3
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	6 + 3
mfn0598	Fisica	6
mfn0600	Elementi di probabilità e statistica	6
mfn0601	Sistemi operativi	6 + 3 + 3
mfn0602	Basi di dati	6 + 3
mfn0603	Linguaggi Formali e Traduttori	6 + 3
mfn0604	Economia e gestione dell'impresa e diritto	9
	Totale 2 anno	**60**

Area tematica di approfondimento Informazione e Conoscenza

L'area tematica di approfondimento "*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. L'area

tematica di approfondimento pone particolare attenzione allo sviluppo di applicazioni desktop e "mobile" di sistemi autonomi e intelligenti, così come alla progettazione di architetture software robuste e flessibili, all'accesso efficiente ai dati e 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 a una preparazione tecnica volta a preparare lo studente/la studentessa per un rapido inserimento nel mondo del lavoro.

III anno

Codice	**Insegnamento**	**CFU**
mfn0605	Programmazione III	4 + 2
mfn0606	Sviluppo delle applicazioni software	4 + 5
mfn0607	Sistemi Intelligenti	6
mfn06	Interazione Uomo Macchina	12

08	e Tecnologie WEB	
mfn1362	Reti I	6
	Crediti liberi a scelta dello studente	**12**
inf0073	Stage	9
inf0074	Prova finale	3
	Totale 3 anno	63

Possibili insegnamenti liberi che rendono il Piano di Studi automaticamente approvato secondo l'area tematica di approfondimento Informazione e Conoscenza.

Codice	**Insegnamento**	**CFU**
mfn0610	Linguaggi e Paradigmi di programmazione **se non si è già scelto mfn1354**	6 + 3
mfn1354	Linguaggi e Paradigmi di programmazione **se non si è già scelto mfn0610**	6
Inf0090	Calcolabilità e Complessità	6

mfn0617	Economia e gestione dell'innovazione	6
mfn0633	Metodi Formali dell'Informatica	9
mfn0636	Sicurezza	6
mfn0618	Sistemi Informativi	6
inf0003	Logica per l'Informatica	6
inf0004	Storia dell'Informatica	6
Inf0193	Metodologie e Tecnologie Didattiche per l'Informatica	6
inf0072	Prolungamento Stage	6

Area tematica di approfondimento Linguaggi e Sistemi

L'area tematica di approfondimento "*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.

Quest'area tematica di approfondimento 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	6 + 3
mfn0633	Metodi Formali dell'Informatica	9
mfn0606	Sviluppo delle applicazioni software	4 + 5
mfn1362	Reti I	6
mfn0605	Programmazione III	4 + 2
	Crediti liberi a scelta dello studente	**12**
inf0073	Stage	9
inf0074	Prova finale	3
	Totale 3 anno	63

Possibili insegnamenti liberi che rendono il Piano di Studi automaticamente approvato secondo l'area tematica di approfondimento Linguaggi e Sistemi:

Codice	**Insegnamento**	**CFU**
Inf0090	Calcolabilità e Complessità	6
mfn0617	Economia e gestione dell'innovazione	6
mfn0608	Interazione Uomo Macchina e tecnologie web **se non si è già scelto mfn1353 o inf0002**	12
mfn1353	Interazione Uomo Macchina **se non si è già scelto mfn0608**	6
inf0002	Servizi Web **se non si è già scelto mfn0608**	6
mfn0636	Sicurezza	6
mfn0618	Sistemi Informativi	6
mfn0607	Sistemi intelligenti	6

mfn0634	Tecnologie web **se non si è già scelto inf0002 o mfn0608**	6
inf0003	Logica per l'Informatica	6
inf0004	Storia dell'Informatica	6
Inf0193	Metodologie e Tecnologie Didattiche per l'Informatica	6
inf0072	Prolungamento Stage	6

Area tematica di approfondimento Reti e Sistemi Informatici

L'obiettivo di questa area tematica di approfondimento 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, tali figure saranno in grado di contribuire alla gestione dell'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	4 + 5
mfn0634	Tecnologie web	6
mfn0635	Reti di elaboratori	12
mfn0636	Sicurezza	6
	Insegnamenti a scelta	**6**
Inf0090	Calcolabilità e Complessità (6 CFU)	6
mfn0605	Programmazione III - (6 CFU)	4 + 2
	Crediti liberi a scelta dello studente	**12**
inf0073	Stage	9
inf0074	Prova finale	3
	Totale 3 anno	63

Possibili insegnamenti liberi che rendono il Piano di Studi automaticamente approvato secondo l'area tematica di approfondimento Reti e Sistemi Informatici:

Codice	**Insegnamento**	**CFU**
mfn0610	Linguaggi e Paradigmi di programmazione **se non si è già scelto mfn1354**	6 + 3
mfn1354	Linguaggi e Paradigmi di programmazione **se non si è già scelto mfn0610**	6
mfn0617	Economia e gestione dell'innovazione	6
mfn1353	Interazione Uomo Macchina	6
mfn0633	Metodi Formali dell'Informatica	9
mfn0618	Sistemi Informativi	6
mfn0607	Sistemi Intelligenti	6
inf0003	Logica per l'Informatica	6
inf00	Storia dell'Informatica	6

04		
Inf0193	Metodologie e Tecnologie Didattiche per l'Informatica	6
inf0072	Prolungamento Stage	6

Struttura dell'offerta formativa rispetto al RAD della classe

Le tabelle che seguono descrivono l'impostazione dell'offerta formativa del nostro corso di Laurea (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 MAT/02 Algebra -- 6 CFU MAT/05 Analisi Matematica -- 9 CFU MAT/09 Ricerca Operativa -- 6 CFU	27	18-30

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 *(minimo 18CFU da decreto ministeriale)*		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	Per la prova finale	3	3-9

la lingua straniera (art. 10, comma 5, lettera d)	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	9	0-15
	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	0-15
Totale Altre Attività		18	18-78

Supporto agli studi

Erasmus

L'Università di Torino ha stabilito accordi bilaterali con un certo numero di Università europee per lo scambio di studenti/studentesse di Informatica nel quadro del progetto Socrates/Erasmus dell'Unione Europea. Gli studenti/le studentesse 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. Dal 16/17, verificata la disponobilità di risorse finanziarie, il Dipartimento ha inoltre previsto un'integrazione di queste borse sulla base dei risultati effettivamente conseguiti.

Gli accordi attualmente esistenti riguardano:

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, Chambéry Francia

Université de Lorraine, Nancy, Francia

Université Paris Diderot – Paris 7, Francia

Technical University of Lodz, Polonia

Universitatea “Politehnica” din Bucuresti, Romania

Universitatea Tehnica Cluj-Napoca, Romania

Université du Luxembourg, Lussemburgo

Universitat Klagenfurt, Austria

Universite Paris Nord – Paris XIII, Francia

Université Pierre et Marie Curie, Paris 6, Francia (per Erasmus Traineeship)

University of Helsinki, Finlandia

Sabaci University, Istanbul, Turchia

Université Montpellier 2, Francia

Wroclaw University of Technology, Polonia

Universidad Politecnica de Valencia, Spagna

Technical University of Sofia, Bulgaria

Universite Jean Monnet de Saint-Etienne, Francia

Universite Paul Sabatier, Toulouse Francia

Panepistimio Thessalias, Volos Grecia

Western Norway University of Applied Sciences, Bergen Norvegia

Rijksuniversiteit Groningen, Olanda

"1 Decembrie 1918" Universitatea Alba Iulia, Romania

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 e le studentesse che per i ricercatori.

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

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

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

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

Supporto on-line agli insegnamenti

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

Aule e laboratori

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

I laboratori didattici (http://di.unito.it/laboratori) offrono oltre 140 postazioni di lavoro (personal computer e workstation) organizzate in cinque aree (laboratorio Turing, laboratorio Von Neumann, laboratorio Dijkstra, laboratorio Babbage e laboratorio Postel). Nei primi quattro si svolgono le lezioni degli insegnamenti di laboratorio, mentre il laboratorio Postel è principalmente dedicato allo sviluppo delle tesi di laurea. Nei periodi in cui i laboratori non sono occupati da lezioni, le postazioni di lavoro sono a

disposizione degli studenti e delle studentesse 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/studentessa 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 e delle studentesse per i soli scopi didattici, e vengono gestiti nella più stretta osservanza delle norme contro la pirateria informatica.

Gli studenti e le studentesse 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 e le studentesse possono utilizzare un'ampia zona studio messa a disposizione dall'Ente Diritto allo Studio Universitario della Regione (EDISU), in locali contigui alle aule. La zona studio è organizzata in una zona aperta in cui gli studenti e le studentesse possono lavorare in gruppo e una zona chiusa in cui è prescritto il silenzio, dedicata allo studio individuale.

Supporto per gli studenti Disabili

L'Università degli Studi di Torino, nella prospettiva di rendere effettivo il diritto allo studio per tutti gli studenti e le studentesse 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 e le studentesse disabili possono usufruire di agevolazioni relative al pagamento delle tasse in relazione alla percentuale di disabilità.

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

- accompagnamento presso le strutture universitarie e gli enti di ricerca e assistenza durante i pasti;
- tutorato didattico: aiuto per la compilazione di appunti, il reperimento testi, fotocopie, disbrigo di pratiche burocratiche;

- supporto per la preparazione degli esami (rivolto esclusivamente a studenti e studentesse con particolari disabilità);
- supporto di interpreti della Lingua Italiana dei Segni e di Mediatori alla Comunicazione per gli studenti e studentesse 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 e le studentesse 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 lo Studio » Studenti con disabilità » 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 e studentesse 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 e studentesse per colloqui specifici e/o prima accoglienza esclusivamente su appuntamento.

La Scuola di Scienze delle Natura, a cui il corso di Laurea in Informatica afferisce, ha nominato come garante per gli studenti e studentesse disabili della propria Scuola la Prof.ssa Anna Capietto, a cui è possibile rivolgersi per informazioni specifiche sui test d'ingresso, sulla modalità d'esame, sui percorsi didattici specifici e particolari.

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

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

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

Tutte le informazioni sono reperibili al seguente link:
- studenti e studentesse con disabilità: http://www.unito.it/servizi/lo-studio/studenti-disabili

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

L'Università di Torino ha uno **sportello dedicato** agli studenti e alle studentesse 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 e le studentesse con DSA. Lo sportello predisposto offre i seguenti servizi:

- interventi di **mediazione** con i/le 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 e le studentesse con DSA è disponibile sul sito di Ateneo www.unito.it » Servizi per lo Studio » Studenti con disabilità » Delegati
In particolare, alla pagina http://www.unito.it/servizi/lo-studio/studenti-con-disabilita/supporto-agli-studenti-con-disabilita-sostenere-gli-esami segnaliamo l'apposito "Modulo di richiesta trattamento individualizzato per esame" che ogni studente/studentessa deve compilare ed inviare via e-mail, con congruo anticipo, al Docente responsabile dell'insegnamento.

Per informazioni

Direzione Risorse Umane

Sportello Dislessia

Via Po, 31 – 10124 Torino

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

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

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

Prof.ssa **Cristina Baroglio**, tel. 0116706711, cristina.baroglio@unito.it, Dipartimento di Informatica, Via Pessinetto 12, Torino

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

Tutte le informazioni sono reperibili al seguente link:

- studenti con DSA: http://www.unito.it/servizi/lo-studio/studenti-con-disturbi-specifici-di-apprendimento-dsa

Indirizzi utili

Ufficio del Corso di Laurea e Laurea Magistrale in Informatica

C.so Svizzera 185, Torino (ingresso da Via Pessinetto 12 – I piano)

*da Lunedì a Venerdì solo su appuntamento**

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

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

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

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

Via Santa Croce, 6 – Torino

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

help desk: https://fire.rettorato.unito.it/helpdesk_ddss/

Telefono: 011 – 670.9900 (numero telefonico unico – selezionare Polo delle Scienze della Natura)

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

Via Michelangelo 32 – Torino

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

e-mail: jp.scienzedellanatura@unito.it

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

Presidente del Consiglio dei Corsi di Laurea e Laurea Magistrale in Informatica:
Prof.ssa Maria Luisa Sapino

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

Direttore del Dipartimento di Informatica: Prof. Luca Console

e-mail: direttore@di.unito.it

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

Vicolo Benevello 3/a, Torino

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

help desk:
https://fire.rettorato.unito.it/helpdesk_ddss/

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

Ente Diritto allo Studio Universitario (http://www.edisu.piemonte.it)
(borse di studio, posto letto, buoni mensa)

Via Giulia di Barolo, 3/bis – 10124 Torino

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

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

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

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

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

e-mail: internationalstudents@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: internationalexchange@unito.it

Telefono: 011 – 670 4425

Rappresentanti degli Studenti: mail rappstudccs@educ.di.unito.it; http://studenti.i-learn.unito.it/

Seconda Parte: Programmi e docenti degli Insegnamenti per l'A.A. 2019/20

In questa seconda parte della guida vengono riportati tutti gli insegnamenti attivati nell'anno accademico 19/20 presso il Corso di Laurea di Informatica, di interesse per gli studenti e le studentesse immatricolati nell'anno e per gli studenti e le studentesse delle due coorti precedenti che devono seguire gli insegnamenti del secondo e terzo anno rispettivamente. Sono inoltre riportati gli insegnamenti che verranno attivati nei prossimi anni, nonché l'elenco dei codici per le convalide (per gli studenti e le studentesse in trasferimento).

Insegnamenti attivi nel 19/20.

La tabella che segue riporta tutti gli insegnamenti che saranno attivi nell'anno accademico 2019/20. Nella tabella le colonne "Codice", "Insegnamento" e "SSD" indicano, rispettivamente, il codice di Ateneo, il titolo dell'insegnamento e il suo settore scientifico disciplinare, TAF e Ambito individuano la tipologia dell'attività formativa (di base, caratterizzante, a scelta dello studente/della studentessa, etc.), "CFU" sono i crediti formativi associati all'insegnamento, "docenti" sono i docenti responsabili dell'insegnamento e "semestre" il periodo didattico in cui si

tiene l'insegnamento. Tutti gli insegnamenti del primo e del secondo anno sono sdoppiati (corso A e corso B)*, mentre i laboratori sono sestuplicati sul 1 anno e quadruplicati sul 2 anno (turni A1, A2, A3, B1, B2 e B3 oppure T1, T2, T3, T4. Ricordiamo che analoghe informazioni sono reperibili sulle pagine web del corso di laurea (da http://laurea.educ.di.unito.it/ seguire il link 'insegnamenti', nella sezione "Per chi studia con noi").

** Corso A: dalla A alla K, Corso B: dalla L alla Z*

Primo anno	INSEGNAMENTO	SSD	TAF	Ambito	CFU	DOCENTI	Semestre	Note
mfn0597	Algoritmi e Strutture Dati	inf/01	B	caratterizzante	9	**Horvath (Teoria A), de' Liguoro (Teoria B), Magro (lab T1), Esposito (lab T2), Pozzato (lab T3), Grosso (lab T4)**	2 sem	
mfn0570	Analisi Matematica	mat/05	A	di base	9	**A: Barutello, Boscaggin B: Dambrosio,**	2 sem	

						Costantini		
mfn 0586	Architettura degli Elaboratori	inf/01	A	di base	9	**Gaeta (teoria A e lab A1), Aldinucci (teoria B), Garetto (lab A2-A3), Lucenteforte (lab B1), Schifanella C (lab B2-B3)**	2 sem	
mfn 0602	Basi di Dati	inf/01	B	caratterizzante	9	**Anselma (Teoria A - lab T1), Basile (lab T2), Pensa (Teoria B – Lab T2-T3-T4), Vernero (lab T4)**	2 sem	
mfn 0612 (inf 0090 dal 19-20)	Calcolabilità e Complessità A (Calcolabilità e	inf/01	D	a scelta	6	**Berardi**	1 sem	

	Complessità dal 19-20)							
mfn0588	Calcolo Matriciale e Ricerca Operativa	mat/09	A	di base	6	**Grosso (A), Aringhieri (B)**	1 sem	
mfn0604	Economia e Gestione dell'Impresa e Diritto	6 CFU sec s-p/08 e 3 CFU ius	C	affine e integrative	9	**Pironti (Economia A e B), Montalcini/Sacchetto (Diritto A e B)**	2 sem	

		/02						
mfn0617	Economia e Gestione dell'Innovazione	sec s-p/08	D	a scelta	6	**Pironti**	1 sem	
mfn0600	Elementi di Probabilità e Statistica	mat/06	C	affine e integrative	6	**Sirovich Roberta (A e B)**	1 sem	
mfn0598	Fisica	fis/01	C	affine e integrative	6	**Migliore (A), Gagliardi (A), Pesando (B)**	2 sem	
mfn1353	Interazione Uomo Macchina	inf/01	D	a scelta	6	**Patti, Segnan**	1 sem	mutua su modulo di Interazione Uomo

								Macchina e Tecnologie Web
mfn0608	Interazione Uomo Macchina e Tecnologie Web	inf/01	B	caratterizzante	12	**Ardissono, Patti, Segnan**	1 sem	
mfn0590	Lingua Inglese I	L-Lin/12	E	conoscenza lingua straniera	3	**Griffin (esercitatore), Radicioni, Bini, Bono, Cordero, Patti (responsabili)**	1 e 2 sem	
mfn0610	Linguaggi e Paradigmi di Programmazi	inf/01	B	caratterizzante	9	**Bono, Padovani**	1 sem	

	one							
mfn 1354	Linguaggi e Paradigmi di Programmazione	inf/01	D	a scelta	6	**Bono, Padovani**	2 sem	mutua su modulo di Linguaggi e Paradigmi di Programmazione da 9 CFU
mfn 0603	Linguaggi Formali e Traduttori	inf/01	B	caratterizzante	9	**Zacchi (prof. a contratto Teoria A), Coppo (prof. a contratto Teoria A), Patti (lab T1), Sproston (lab T2), Padovani (Teoria B), Patti (lab T3), Sproston (lab T4), tutoraggio A e B Di Caro**	1 sem	

inf0003	Logica per l'Informatica	mat/01	D	a scelta	6	**Paolini**	1 e 2 sem	
mfn0578	Matematica Discreta e logica	6 CFU mat/02 + 6 CFU mat/01	A	di base	12	**Murru (Mat Discr A), Andretta (Log A)** **Mori (Mat Discr B), Motto Ros (Log B)**	1 sem	
mfn0633	Metodi Formali dell'Informatica	inf/01	B	caratterizzante	9	**de' Liguoro**	1 e 2 sem	
mfn	Progra	inf	A	di	9	**Cardone (teoria**	1	

0582	mmazione I	/01		base		A), tutoraggio A Ruffo, Gliozzi (lab A3-B2-B3) Mazzei (lab A2), Roversi (Teoria B, lab B1), tutoraggio B Rapp, Cardone-Amparore (lab B2)	sem	
mfn0585	Programmazione II	inf/01	A	di base	9	Bono (teoria A), tutoraggio A Amparore, Berardi (teoria B – lab B3), tutoraggio B Schifanella R., Damiani (lab A1-A2), Amparore (lab A3), Torta (lab B1), Magro (lab B2)	2 sem	
mfn060	Programmazi	inf/0	B	caratterizza	6	Ardissono	1 se	

5	one III	1		nte			m	
mfn 063 5	Reti di Elaboratori	inf /0 1	B	caratterizzante	12	**Garetto, Sereno**	1 e 2 sem	
mfn 136 2	Reti I	inf /0 1	B	caratterizzante	6	**Botta**	1 sem	
inf0 002	Servizi Web	inf /0 1	D	a scelta	6	**Ardissono**	1 sem	mutua su modulo di Interazione Uomo Macchina e Tecnologie Web
mfn 063 6	Sicurezza	inf /0 1	B	caratterizzante	6	**Bergadano**	2 sem	
mfn 061 8	Sistemi Inform	inf /0 1	D	a scelta	6	**Micalizio**	1	

	ativi							
mfn 0607	Sistemi Intelligenti	inf/01	B	caratterizzante	6	**Baroglio**	2 sem	
mfn 0601	Sistemi Operativi	inf/01	B	caratterizzante	12	**Gunetti (Teoria A), Baroglio (Teoria B), De Pierro (lab T1 C), Radicioni (lab T1 Unix), Bini (lab T2 C e T2 Unix; lab T3 C e T3 Unix), De Pierro (lab T4 C), Schifanella C (lab T4 Unix)**	1 sem	
mfn 0606	Sviluppo delle Applicazioni Software	inf/01	B	caratterizzante	9	**Baldoni (Teoria), Capecchi (lab 1), Picardi (lab 2)**	2 sem	

inf0004	Storia dell'Informatica	inf/01	D	a scelta	6	**Gunetti, Cardone**	1 e 2 sem	
mfn0634	Tecnologie Web	inf/01	B	caratterizzante	6	**Ruffo, Botta**	1 sem	
Inf0073	Stage		F	altre attività	9			
Inf0072	Prolungamento stage		D	a scelta	6			
Inf0074	Prova Finale		E	altre attività	3			

Codici per convalide di insegnamenti e competenze.

Gli studenti e le studentesse che per effetto di passaggio o trasferimento abbiano ottenuto la convalida di crediti liberi con nessuna corrispondenza specifica a insegnamenti del Corso di

Laurea in 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 insegnamento libero da selezionare dalla lista prevista per il percorso scelto, in caso non sia ancora stato raggiunto il minimo di 12 CFU.

mfn1409	Altre Attività	altre attività, a scelta lettera A	12
mfn1408	Altre Attività	altre attività, a scelta lettera A	6
mfn1522	Altre Attività	altre attività, a scelta lettera A	6

Informazioni aggiornate al 15 ottobre 2019

Programmi e altre informazioni per gli insegnamenti attivi nel 19/20 (syllabus degli insegnamenti)

Questa parte sarà disponibile a ottobre, come da regolamento didattico di Ateneo. Gli studenti e le studentesse possono intanto consultare le pagine degli insegnamenti del 19/20 sul sito web del corso di studio, che sono in fase di ultimazione in questi giorni.

Insegnamento	**MFN0597 - Algoritmi e Strutture Dati**
Insegnamento (inglese):	**Algorithms and Data Structures**
CFU:	**9**
Settore:	**INF/01 - INFORMATICA**
Periodo didattico:	**2**
Tipologia di Attività Formativa:	**B - caratterizzante**
Docenti:	**Ugo DE' LIGUORO (Titolare)** **Roberto ESPOSITO (Titolare)** **Andrea Cesare GROSSO (Titolare)** **Andras HORVATH (Titolare)**

Diego MAGRO (Titolare)

Gian Luca POZZATO (Titolare)

---

1. Prerequisiti e Propedeuticità:

Competenze attese in ingresso

Si presuppone che lo studente sia a conoscenza delle basi della programmazione e dei linguaggi di programmazione Java e C; che sia in possesso delle nozioni elementari della matematica del discreto, del continuo, e della logica matematica.

Eventuali corsi propedeutici

Programmazione I & Laboratorio, Programmazione II & Laboratorio, Matematica Discreta, Analisi Matematica, Logica Matematica, Sistemi Operativi (in particolare l'apprendimento del linguaggio C).

2. Obiettivi formativi:

L'insegnamento ha lo scopo di introdurre i concetti e le tecniche fondamentali per l'analisi e la progettazione di

algoritmi, che sono alla base dello sviluppo del software. Gli studenti acquisiranno conoscenze circa l'analisi di correttezza e complessità computazionale degli algoritmi, sulle strutture dati per la rappresentazione dell'informazione, sulle tecniche di problem solving mediante lo sviluppo di algoritmi efficienti. L'insegnamento è supportato da un laboratorio che ne costituisce parte integrante, finalizzato alla realizzazione e sperimentazione degli algoritmi e delle strutture dati mediante un linguaggio imperativo ed uno object-oriented.

3. Risultati dell'apprendimento attesi:

Al termine del corso ci si aspetta che lo studente sia in grado di risolvere problemi algoritmici elementari mediante la progettazione degli algoritmi e l'analisi delle soluzioni adottate, sia per quanto riguarda l'efficienza, sia per quanto concerne la scelta delle strutture dati manipolate dagli algoritmi. Lo studente sarà in grado di individuare e comprendere algoritmi illustrati nella letteratura specialistica e di adattarli realizzando concretamente le proprie implementazioni, tanto producendo il codice dei programmi quanto

utilizzando librerie software preesistenti. Infine lo studente dovrà essere in grado di comunicare e documentare le soluzioni adottate in modo tecnicamente rigoroso.

4. Modalità di verifica dell'apprendimento:

L'esame di Algoritmi e strutture dati consiste in una prova scritta ed in una discussione orale del progetto di laboratorio.

ll progetto di laboratorio va consegnato entro e non oltre la data della prova scritta che si intende sostenere. E' vietato sostenere la prova scritta in caso di mancata consegna del progetto di laboratorio.

Il superamento della prova scritta è condizione necessaria per poter sostenere la discussione del progetto di laboratorio, la quale deve essere sostenuta nella stessa sessione in cui è stato superato lo scritto (si ricorda che le sessioni sono quattro: 1. giugno-luglio, 2. settembre, 3. dicembre e 4. gennaio-febbraio). Nel caso in cui questa seconda prova non venga superata entro i termini della sessione, lo scritto dovrà essere

ripetuto. Per sostenere la prova orale, oltre all'iscrizione ufficiale all'appello, è necessaria una prenotazione mediante apposita procedura che sarà messa a disposizione sulla pagina i-learn del corso.
Il voto sarà la media pesata (sui rispettivi crediti) dei voti ottenuti nelle due prove scritta ed orale, valutate in 30+1 esimi, essendo comunque necessario il raggiungimento della sufficienza in entrambe le prove.

5. Modalità d'insegnamento:

Sono previste 48 ore di lezione frontale ed esercitazioni in aula, e 24 ore di attività guidata dai docenti in laboratorio. Per le indicazioni bibliografiche, la distribuzione di note ed altro materiale didattico, e per la consegna e valutazione degli elaborati delle esercitazioni si farà uso della piattaforma e-learning Moodle, cui si richiede agli studenti l'iscrizione.

6. Attività di supporto:

Il materiale didattico sarà reso via via disponibile on line attraverso la piattaforma Moodle.

7. Programma:

Programma

- Problemi e algoritmi: risolubilità, correttezza, complessità.
- Analisi computazionale e complessità asintotica
- Algoritmi di ordinamento
  - Algoritmi elementari quadratici
  - Divide et impera: mergesort e quicksort
  - Risoluzione di relazioni di ricorrenza
  - Limiti inferiori per l'ordinamento
- Programmazione dinamica
- Massima sottosequenza comune
- Strutture dati
- Strutture concrete: array, liste, tabelle hash
- Strutture astratte: pile, code, dizionari
- Code di priorità, heapsort
- Analisi ammortizzata, cenni
- Alberi
  - Definizione e visita

Alberi di ricerca

Alberi rosso-neri

- Grafi

Definizione e visita

Ordinamento topologico e componenti fortemente connesse

Algoritmi greedy: alberi di copertura minima

Cammini minimi: algoritmo di Dijkstra

8. Testi consigliati e bibliografia:

1. Per la teoria:
   - Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein: Introduzione agli algoritmi e strutture dati, McGraw-Hill 2010.
2. Per il laboratorio:
   - Horstmann: Concetti di informatica e fondamenti di Java, Quinta Edizione (o successiva), Apogeo, 2010.

Insegnamento

MFN0570 - Analisi Matematica

Insegnamento (inglese):	**Mathematical Analyis**
CFU:	**9**
Settore:	**MAT/05 - ANALISI MATEMATICA**
Periodo didattico:	**2**
Tipologia di Attività Formativa:	**A - di base**
Docenti:	**Vivina Laura BARUTELLO (Titolare)** **Alberto BOSCAGGIN (Titolare)** **Camillo COSTANTINI (Titolare)** **Walter DAMBROSIO (Titolare)**

1. Prerequisiti e Propedeuticità:

Competenze attese in ingresso

L'insegnamento prevede la conoscenza dei contenuti di matematica di base forniti dalla scuola secondaria di secondo grado.

In particolare, a livello di conoscenze e comprensione in ingresso lo studente dovrà:

â•• conoscere i concetti di base sulla retta, sia dal punto di vista della geometria sintetica sia della geometria analitica, con particolare riferimento al concetto di pendenza; â•• conoscere le funzioni quadratiche e le loro proprietà algebriche e grafiche;

â•• ricordare le proprietà delle potenze e dei logaritmi e conoscere i grafici delle funzioni potenza, esponenziali e logaritmiche;

â•• conoscere gli elementi essenziali di trigonometria (misure degli angoli in radianti, grafici delle funzioni circolari);

â•• conoscere i concetti di dominio, immagine, zeri, segno e monotonia per funzioni reali di una variabile reale.

Inoltre, come applicazione di conoscenza e comprensione, lo studente dovrà saper:

â •• determinare l'equazione della retta passante per un punto ed avente pendenza assegnata e l'equazione della retta passante per due punti;

â •• determinare l'equazione di una retta a partire dal suo grafico (calcolo di pendenza e intercetta);

â •• tracciare il grafico di una funzione lineare e determinare per via grafica dominio, immagine, zeri e segno, monotonia;

â •• tracciare il grafico di una funzione quadratica e determinare per via grafica dominio, immagine, zeri e segno, monotonia;

â •• riconoscere come varia la retta tangente al grafico di una funzione quadratica in un suo punto, anche in relazione alla concavità della funzione; â •• risolvere equazioni e disequazioni di primo e secondo grado, anche per via grafica;

â •• tracciare il grafico di funzioni potenza xa, con a intero positivo o negativo, e determinarne per via grafica dominio, immagine, simmetrie, zeri e segno, monotonia, massimi e minimi;

â•• tracciare il grafico di funzioni del tipo a^x o log_a x, con a positivo, e determinarne per via grafica dominio, immagine, zeri e segno, monotonia;

â•• risolvere equazioni e disequazioni del tipo a^x=b, a^x>b, a^x <b, log_a x=b, log_a x>b e log_a x<b;
â•• trasformare la misura di un angolo da gradi a radianti e viceversa; â•• tracciare il grafico delle funzioni circolari; â•• risolvere equazioni del tipo sin x =b e cos x=b;

â•• determinare dominio, immagine, zeri e segno, monotonia di una funzione a partire dal suo grafico.

Eventuali corsi propedeutici

I prerequisiti richiesti potranno essere recuperati attraverso il Corso di Riallineamento di Matematica presente sulla Piattaforma Orient@mente.

2. Obiettivi formativi:

L'insegnamento ha lo scopo di presentare le nozioni di base su funzioni, grafici e loro trasformazioni, di introdurre i concetti di derivata e di integrale definito e di illustrare l'utilizzo di tecniche di tipo analitico nello studio di fenomeni discreti. Si tratta di argomenti indispensabili per la formazione dei laureati in scienze dell'informazione (classe L-31).

L'insegnamento concorre agli obiettivi formativi dell'area Matematico-Fisica del corso di Laurea in Informatica, fornendo conoscenze relative ai concetti ed agli strumenti matematici e metodologici fondamentali necessari per descrivere, schematizzare e interpretare i principali aspetti della realtà che ci circonda. In particolare, l'insegnamento si inoltre propone di accrescere le capacità di comprensione degli studenti e di consentire loro di acquisire un modo rigoroso ed analitico di ragionare e affrontare nuovi problemi.

La significativa presenza di teoremi, molti dei quali con dimostrazione, ha l'obiettivo di rafforzare nello studente le attitudini logico-deduttive apprese nel corso di Matematica discreta e Logica.

3. Risultati dell'apprendimento attesi:

Conoscenza e comprensione

Alla fine di questo insegnamento lo studente saprà: â•• riconoscere i grafici e le proprietà asintotiche delle funzioni elementari;
â•• rievocare e ricordare la definizione di funzione monotona e spiegare e descrivere il legame tra la monotonia di una funzione composta e quella delle funzioni componenti;

â•• spiegare e illustrare il significato intuitivo di limite, anche in riferimento alla sua interpretazione in termini grafici;

â•• riconoscere la definizione di limite, sia nel caso continuo (funzioni), sia nel caso discreto (successioni); â•• ricordare e rievocare la definizione ed il significato di asintoto verticale o orizzontale;
â•• enunciare e dimostrare alcuni risultati fondamentali sui limiti, illustrandone il significato anche a livello grafico;

â•• riconoscere e ricordare la definizione di funzione continua ed illustrare criticamente il suo legame con il concetto di limite; â•• enunciare i principali risultati relativi alle funzioni continue;

â•• confrontare le diverse crescite di una funzione all'infinito e distinguere tra crescita logaritmica, polinomiale ed esponenziale, sia in termini grafici sia in termini quantitativi;

â•• ricordare le definizioni dei simboli di Landau e spiegare le relazioni che intercorrono tra i vari simboli;

â•• ricordare la definizione di successione geometrica, in termini ricorsivi o iterativi, e illustrarne le principali applicazioni;

â•• discutere il comportamento asintotico e le proprietà qualitative di una successione geometrica in dipendenza dai valori della sua ragione; â•• riconoscere una successione definita per ricorrenza; â•• discutere la stabilità dell'equilibrio di una successione per ricorrenza lineare del primo ordine;

â•• spiegare l'utilizzo dei corretti simboli di Landau nella discussione della complessità di un algoritmo e determinare tale complessità in alcuni semplici casi;
â•• riconoscere e ricordare la definizione di derivata di una funzione in un punto;

â•• interpretare il concetto di derivata di una funzione in un punto da vari punti di vista applicativi

(pendenza, velocità istantanea, tasso istantaneo di variazione); â•• ricordare l'espressione della retta tangente al grafico di una funzione in un punto;
â•• illustrare il legame tra funzioni continue e funzioni derivabili, fornendo ove necessario dimostrazioni, esempi e controesempi;

â•• rievocare e ricordare l'espressione della derivata delle funzioni elementari e le regole di derivazione;

â•• ricordare la definizione di primitiva di una funzione ed illustrare il legame tra diverse primitive della stessa funzione su un intervallo;

â•• enunciare il Teorema di Lagrange ed illustrarne le sue interpretazioni cinematica e geometrica e le sue principali conseguenze (legame tra la monotonia di una funzione ed il segno della sua derivata, caratterizzazione delle funzioni a derivata ovunque nulla su un intervallo, legame tra diverse primitive della stessa funzione su un intervallo);

â•• collegare le proprietà geometriche di concavità/convessità del grafico di una funzione con le proprietà di segno e monotonia delle derivate della

funzione; â•• dimostrare i principali risultati relativi al calcolo differenziale; â•• spiegare il problema dell'approssimazione locale di una funzione in un punto;

â•• ricordare e riconoscere l'espressione dei polinomi di Taylor di una funzione in un punto, valutando in termini anche quantitativi l'errore commesso approssimando una funzione con un suo polinomio di Taylor; â•• ricordare i polinomi di Maclaurin delle funzioni elementari; â•• illustrare il problema della risolubilità esatta ed approssimata di un'equazione; â•• enunciare e dimostrare il Teorema di esistenza degli zeri;

â•• descrivere il metodo di Newton per la risoluzione approssimata di un'equazione, discutendone le proprietà di convergenza;

â•• enunciare e dimostrare il Teorema sulla convergenza del metodo di Newton per la risoluzione approssimata di un'equazione;

â•• confrontare il metodo di bisezione ed il metodo di Newton per la risoluzione approssimata di un'equazione, evidenziando gli aspetti vantaggiosi in ognuno di essi; â•• descrivere la definizione di integrale definito di una funzione su un intervallo;

â•• interpretare la definizione di integrale definito in termini di lavoro, spostamento netto e valor medio; â•• collegare il concetto di integrale definito con quello di area di regioni piane;

â•• ricordare la formula del punto medio per il calcolo approssimato di un integrale definito e valutare l'errore commesso nell'approssimazione; â•• rievocare e spiegare le proprietà di linearità e monotonia dell'integrale definito;

â•• enunciare il Teorema fondamentale del calcolo integrale ed interpretarlo criticamente, evidenziando la sua centralità rispetto alle nozioni di derivata e integrale definito;

â•• enunciare il Teorema di Torricelli-Barrow e illustrare le sue conseguenze sul calcolo esatto di un integrale definito; â•• dimostrare i principali risultati sul calcolo integrale;

â•• spiegare il concetto di integrale improprio, in

relazione al problema dell'area di una regione illimitata di piano; â•• enunciare e dimostrare i principali risultati relativi alla convergenza di un integrale improprio;

â•• dimostrare la convergenza dell'integrale improprio della funzione gaussiana e spiegarne la sua importanza in statistica; â•• spiegare il concetto di serie ed illustrarne il legame con il concetto di successione; â•• ricordare la definizione di serie convergente, divergente o indeterminata; â•• ricordare la serie geometrica e discuterne la convergenza in dipendenza dalla ragione; â•• discutere alcune applicazioni della serie geometrica; â•• ricordare le serie armoniche generalizzate e discuterne convergenza e divergenza; â•• enunciare e dimostrare i principali risultati teorici sulla convergenza di una serie;
â•• spiegare il legame tra integrali impropri e serie, enunciando e dimostrando il teorema che li mette in relazione, sotto opportune ipotesi.

Applicare conoscenza e comprensione

Alla fine di questo insegnamento lo studente avrà sviluppato capacità di lavorare sia su aspetti grafici sia

su aspetti di calcolo, approssimato o esatto.

In particolare, a livello di grafici saprà:

â•• dedurre dal grafico di una funzione informazioni qualitative e quantitative sulla funzione stessa (dominio, immagine, monotonia, zeri, segno, limiti);

â•• ottenere dal grafico di una funzione il grafico di nuove funzioni, mediante trasformazioni geometriche o mediante l'uso delle proprietà delle funzioni composte; â•• tracciare il grafico di funzioni ottenute da funzioni elementari mediante composizioni; â•• stimare il valore della derivata della funzione in un punto a partire dal grafico di una funzione;

â•• tracciare il grafico della derivata di una funzione a partire dal grafico della funzione stessa, analizzando in modo critico i legami tra una funzione e la sua derivata;

â•• interpretare il grafico della numerosità di una popolazione in funzione del tempo e tracciare da esso il grafico del tasso di crescita in funzione del tempo;

â•• tracciare il grafico della velocità di un oggetto che si muove di moto rettilineo, a partire dal grafico della sua posizione;

â•• tracciare il grafico della posizione di un oggetto che si muove di moto rettilineo, a partire dal grafico della sua velocità e dalla posizione all'istante iniziale;
â•• discutere la risolubilità di un'equazione, mediante il confronto tra grafici.

A livello di calcolo approssimato saprà:

â•• determinare l'approssimazione di una funzione in un punto mediante un polinomio di Taylor e utilizzarla per stimare i valori della funzione;

â•• stimare l'errore commesso nell'approssimazione locale di una funzione mediante un polinomio di Taylor;

â•• eseguire ed implementare l'algoritmo di bisezione e

lo schema ricorsivo del metodo di Newton per stimare le soluzioni di un'equazione;

â•• calcolare in modo approssimato integrali definiti utilizzando la formula del punto medio, sia a partire dall'espressione esplicita della funzione integranda, sia a partire dal suo grafico;

â•• stimare l'errore commesso nel calcolo approssimato di un integrale definito mediante la formula del punto medio.

A livello di calcolo esatto saprà: â•• determinare l'espressione di successioni per ricorrenza del primo ordine lineari; â•• calcolare la derivata di una funzione; â•• determinare l'approssimazione di una funzione in un punto mediante un polinomio di Taylor; â•• determinare la retta tangente al grafico di una funzione in un suo punto; â•• calcolare le primitive di funzioni, in casi immediati; â•• calcolare integrali definiti immediati; â•• determinare la somma di una serie geometrica, anche in funzione di eventuali parametri.
Inoltre, lo studente saprà:

â•• discutere la convergenza di un integrale improprio o

di una serie, utilizzando in modo opportuno i criteri di convergenza;

â•• interpretare e rielaborare grafici qualitativi e dati quantitativi di fenomeni di tipo fisico (cinematica del punto) o biologico (popolazioni e loro dinamica).

Autonomia di giudizio

Alla fine di questo insegnamento lo studente saprà:

â•• riconoscere ed individuare argomentazioni logiche con una chiara identificazione di assunti e conclusioni;
â•• collegare e commentare criticamente i principali risultati teorici illustrati nel corso

dell'insegnamento, individuando i legami che tra essi intercorrono.

4. Modalità di verifica dell'apprendimento:

L'esame è strutturato in tre prove scritte, obbligatorie per tutti gli studenti, indipendentemente dall'Anno Accademico di iscrizione, tutte da sostenere nello stesso appello. Le prove si svolgono in successione: per accedere alla seconda è necessario aver superato la prima, per accedere alla terza aver superato la seconda.

Prima prova (quiz) La prima prova consiste nella risposta a cinque domande a scelta multipla che hanno l'obiettivo di verificare la capacità dello studente di risolvere semplici esercizi su argomenti di base. La durata è di venti minuti; per superare il test occorre rispondere in modo corretto ad almeno 4 domande su 5. L'esito è: superato o non superato ed è noto immediatamente al termine del test stesso; chi non supera il test non può accedere alla prova d'esame.

Seconda prova (teoria) La seconda prova verte sugli argomenti trattati a lezione ed esercitazioni e mira a valutare le conoscenze di definizioni e enunciati di

teoremi, nonché le capacità logicodeduttive nell'esposizione delle dimostrazioni presentate a lezione. La prova è valutata in trentesimi; si accede alla terza prova con una valutazione almeno pari a 15/30.

Terza prova (esercizi) La terza prova prevede la risoluzione di un certo numero di esercizi, in cui lo studente deve dimostrare di saper applicare le sue conoscenze in problemi maggiormente strutturati. La prova è valutata in trentesimi ed è superata con una valutazione almeno pari a 18/30.

L'intero esame è superato se la media tra i risultati della seconda e terza prova è almeno pari a 18/30.

Modalità di iscrizione e regole per lo svolgimento degli esami La prenotazione agli appelli d'esame tramite Esse3 ed entro una settimana dalla data dell'esame è obbligatoria e indispensabile. Non si accetteranno prenotazioni pervenute via mail e non verranno ammessi studenti che non si siano prenotati. Inoltre, per una migliore organizzazione dei Laboratori informatici, chi si prenota e non si presenta all'esame senza prima avvisare i docenti non avrà diritto a partecipare all'appello successivo. Per sostenere l'esame è necessario presentarsi con un documento di

riconoscimento (preferibilmente la smartcard) e ricordare le credenziali di Ateneo (username e password), che dovranno essere digitate sul computer dell'aula per iniziare le prove. Durante l'esame non è consentito l'uso di strumenti elettronici e non è permesso consultare testi o appunti. Durante la terza prova si può utilizzare la calcolatrice disponibile sul computer. E' assolutamente vietato, pena l'esclusione dall'esame, tenere alla postazione informatica telefoni cellulari, tablet e simili (anche se spenti, in tasca,..). La presenza di uno di questi apparecchi, anche spento, comporterà l'espulsione immediata dall'aula e l'annullamento della prova. Informazioni per gli studenti degli anni accademici passati

Gli studenti degli anni accademici passati devono sostenere l'esame secondo le modalità dell'anno accademico corrente.

5. Modalità d'insegnamento:

Le modalità di insegnamento comprendono: lezioni frontali, lezioni inverse (flipped), apprendimento attivo in aula e a distanza, esercitazioni in aula.

â•• Lezioni frontali e attività in aula: lezioni frontali eventualmente supportate dall'uso di strumenti di videoscrittura e di software di visualizzazione dinamica; attività ed esercitazioni in aula con eventuale partecipazione degli studenti (svolgimento di esercizi, discussioni, gruppi di lavoro).

â•• Attività e materiale online (Piattaforma Moodle): calendario delle lezioni e delle esercitazioni; video sostitutivi delle lezioni frontali per argomenti erogati in modalità inversa (flipped); quiz ed assegnazioni per l'apprendimento e l'autovalutazione.

L'insegnamento, con le sue modalità ed attività, contribuisce a formare e consolidare le seguenti competenze trasversali: gestione del tempo, attraverso lo svolgimento di prove di autovalutazione informatizzate aventi tempo stabilito; corretta attribuzione causale di successi ed insuccessi,

attraverso lo svolgimento di prove di autovalutazione con feedback da parte del tutor o dei docenti.

6. Attività di supporto:

Sono disponibili settimanalmente su piattaforma Moodle i Fogli di esercizi, in cui vengono proposti esercizi simili a quelli affrontati durante le esercitazioni. Questi esercizi possono essere svolti a casa oppure, in modo individuale o a piccoli gruppi, durante gli incontri settimanali del Tutorato disciplinare, in cui uno studente della LT in Matematica è a disposizione per aiutare nella loro risoluzione e per fornire chiarimenti e spiegazioni. Gli esercizi proposti svolgono la funzione di valutazione formativa in itinere ed hanno un ruolo assai importante nello studio degli argomenti del corso: il loro svolgimento e la loro correzione, effettuata negli incontri del tutorato, consentono allo studente di verificare costantemente la propria preparazione ed il grado di autonomia nel risolvere problemi simili a quelli affrontati durante le esercitazioni e della stessa difficoltà di quelli presenti all'esame.

Sono inoltre disponibili su piattaforma Moodle prove di autovalutazione e simulazioni delle prove d'esame.

7. Programma:

1. Funzioni, grafici e modelli

1.1. Funzioni elementari e loro grafici

1.2. Trasformazioni geometriche di grafici

1.3. Grafici di funzioni composte

2. Il concetto di limite

2.1. Il concetto di limite nel caso continuo (limite di funzioni)

2.2. Il concetto di limite nel caso discreto (limite di successioni)

2.3. Principali risultati teorici sui limiti

2.4. Successioni definite per ricorrenza

2.5. Crescite e confronti di crescite: i simboli di Landau

3. Calcolo differenziale

3.1. Derivata di una funzione in un punto

3.2. Funzione derivata e funzioni primitive; relazioni tra una funzione e la sua derivata o le sueprimitive

3.3. Derivata e monotonia; derivata e convessità

3.4. Approssimazione locale di funzioni mediante polinomi

4. Risoluzione approssimata di equazioni

4.1. Il Teorema di esistenza degli zeri ed il metodo di bisezione

4.2. Il metodo di Newton

5. Calcolo integrale

5.1. Integrale definito di una funzione su un intervallo

5.2. Teorema Fondamentale del Calcolo Integrale

5.3. Teorema di Torricelli-Barrow

5.4. Integrali impropri

6. Serie numeriche

6.1. La serie geometrica

6.2. Le serie armoniche generalizzate

6.3. Definizioni e risultati teorici sulle serie

6.4. Confronto tra serie ed integrali impropri

8. Testi consigliati e bibliografia:

M. Bramanti, C.D. Pagani, S. Salsa "Analisi Matematica 1" © Zanichelli Editore

W. Dambrosio "Analisi Matematica" © Zanichelli Editore

Insegnamento	MFN0586 - Architettura degli Elaboratori

Insegnamento (inglese):	Computer architecture
CFU:	9
Settore:	INF/01 - INFORMATICA
Periodo didattico:	2
Tipologia di Attività Formativa:	A - di base
Docenti:	Marco ALDINUCCI (Titolare) Rossano GAETA (Titolare) Michele GARETTO (Titolare) Maurizio LUCENTEFORTE (Titolare) Claudio SCHIFANELLA (Titolare)

1. Prerequisiti e Propedeuticità:

Competenze attese in ingresso

Conoscenza di base del linguaggio Java, nozioni di programmazione, Algebra di Boole.

Eventuali corsi propedeutici

È richiesta la conoscenza delle competenze fornite dai corsi di:

- Programmazione I e Laboratorio (I semestre)
- Matematica Discreta e Logica (I semestre)

2. Obiettivi formativi:

L'insegnamento si propone di fornire allo studente:

- la conoscenza delle tecniche di codifica dell'informazione all'interno dei calcolatori;
- la conoscenza dell'organizzazione hardware degli elaboratori, attraverso la nozione di gerarchiedi macchine virtuali;
- la comprensione delle funzioni svolte dall'hardware e utilizzate dai sistemi operativi;
- le nozioni di base per la comprensione del processo di traduzione da linguaggi ad alto livello alinguaggio macchina;
- le conoscenze delle tecniche di microprogrammazione e di programmazione in un linguaggioassemblativo.

3. Risultati dell'apprendimento attesi:

Al termine dell'insegnamento lo studente dovrà dimostrare di essere in grado di:

- codificare e decodificare numeri con e senza segno, interi, frazionari, in virgola mobile;
- conoscere e analizzare circuiti logici elementari;
- saper analizzare i componenti digitali che sono alla base dell'architettura dei moderni calcolatori;-
- conoscere e saper utilizzare le tecniche della microprogrammazione per l'esecuzione di istruzioni a livello macchina;
- conoscere le relazioni tra linguaggi ad alto livello (Java) ed equivalente linguaggio a livellomacchina;
- saper sviluppare piccoli programmi in un linguaggio assemblativo e conoscere le relativetecniche di traduzione in linguaggio macchina e del collegamento (linking/loading).

4. Modalità di verifica dell'apprendimento:

L'esame consiste di una prova scritta con voto finale espresso in trentesimi (con possibilità di lode). La prova

scritta è articolata in due parti: una relativa agli argomenti di teoria ed una relativa agli argomenti di laboratorio. Ognuna delle due parti prevede un punteggio espresso in trentesimi ed il voto finale è definito come la media (ponderata rispetto ai CFU della teoria e del laboratorio) della valutazione delle due componenti. Le domande di teoria possono essere a risposta aperta oppure a risposta chiusa. Le domande a risposta aperta possono richiedere l'enunciazione e la descrizione di proprietà o di tecniche viste a lezione. Possono altresì richiedere l'applicazione della teoria a particolari esempi. Con queste domande si intende valutare la comprensione dei principi visti a lezione, la capacità di enunciarli con l'uso appropriato di un linguaggio scientifico e la capacità di applicarli a casi pratici. Le domande di laboratorio vertono sulla microprogrammazione e sulla programmazione in livello assemblativo.

5. Modalità d'insegnamento:

L'insegnamento è diviso in una parte di teoria e una di laboratorio. Per la parte di teoria sono previste 48 ore lezioni frontali, integrate da esempi e da esercitazioni. Per la parte di laboratorio sono previste 30 ore di attività in laboratorio in cui, con l'ausilio di appositi simulatori e applicativi, si svolgono esercizi e approfondimenti sugli argomenti del corso. La frequenza è facoltativa, consigliata, e la prova finale sarà uguale per frequentanti e non.

6. Attività di supporto:

Il materiale didattico di supporto sia per il corso A che per il corso B (lucidi, link, esempi di testo di esami ed altro) è disponibile presso il supporto on-line ai corsi I-learn http://i-learn.educ.di.unito.it/.

Testi e soluzioni di alcuni esercizi svolti in aula possono essere trovati nelle pagine del corso di Architettura degli Elaboratori per gli anni accademici precedenti.

7. Programma:

-Introduzione all'organizzazione strutturata dei calcolatori: macchina di Von Neumann, macchine virtuali;
-Codifica dell'informazione: numeri binari, conversione tra basi, numeri negativi, operazioni tranumeri binari, numeri floating point e standard IEEE 754;
-Livello logico digitale: porte logiche e algebra di Boole, circuiti logici, latch, flip-flop, registri, chipdi memoria, RAM e ROM;
-Memoria e organizzazione della memoria: organizzazione della memoria, memoria cache,memorie permanenti;
-Il bus;
-Il livello della microarchitettura: organizzazione della CPU ed esecuzione delle istruzioni, esempiodi microarchitettura, l'ISA IJVM, il microprogramma per l'architettura Mic-1, ottimizzazione dell'architettura Mic-1 (Mic-2) ed un'architettura con pipeline (Mic-3);
-Il livello ISA: sommario al livello ISA, caratteristiche, modelli di memoria, registri e istruzioni,modalità di indirizzamento, formati delle istruzioni, tipi di dati ed istruzioni nel livello ISA, controllo del flusso, I/O,

interrupt, trap; - Assembler, linker, loader;

In laboratorio saranno svolte esercitazioni sulle codifiche, sui circuiti logici, sulla microprogrammazione in Mic-1, sulla scrittura di programmi in linguaggio assemblativo IJVM, con l'ausilio di appositi simulatori e applicativi.

8. Testi consigliati e bibliografia:

Andrew S. Tanenbaum, Todd Austin Architettura dei calcolatori: un approccio strutturale. 6a edizione Pearson Education Italia, 2013. ISBN 9788871929620 oppure

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

Insegnamento	**MFN0602 - Basi di Dati**
Insegnamento (inglese):	**Databases**

CFU:	**9**
Settore:	**INF/01 - INFORMATICA**
Periodo didattico:	**2**
Tipologia di Attività Formativa:	**B - caratterizzante**
Docenti:	**Luca ANSELMA (Titolare)** **Valerio BASILE (Titolare)** **Ruggero Gaetano PENSA (Titolare)** **Fabiana VERNERO (Titolare)**

---

1. Prerequisiti e Propedeuticità:

Competenze attese in ingresso

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

Eventuali corsi propedeutici

Insegnamenti di Logica, Matematica Discreta e Programmazione 1 e 2, Sistemi Operativi.

2. Obiettivi formativi:

L'insegnamento concorre al raggiungimento degli obiettivi formativi del Corso di Laurea in

Informatica fornendo un'introduzione alle basi di dati e ai sistemi di gestione delle medesime (DBMS). Si propone perciò di fornire le competenze teoriche e pratiche necessarie alla progettazione e gestione delle basi di dati, e la padronanza degli strumenti per analizzare, progettare, sviluppare e gestire le basi di dati.

3. Risultati dell'apprendimento attesi:

Conoscenza e capacità di comprensione

Al termine dell'insegnamento lo studente dovrà conoscere:

- i fondamenti matematici del modello relazionale,
- i linguaggi per le interrogazioni in algebra ed in calcolo relazionale e in SQL, i fondamenti della teoria della normalizzazione,

i concetti e l'uso di transazioni, le strutture dati per la memorizzazione efficiente dell'informazione.

Capacità di applicare conoscenza e comprensione

Al termine dell'insegnamento lo studente dovrà essere in grado di:

- analizzare i requisiti per la progettazione di una base di dati,

rappresentare i requisiti in forma di diagrammi Entity Relationship (ER), trasformare schemi ER in schemi relazionali, trasformare interrogazioni in forma testuale in interrogazioni SQL, eseguire semplici ristrutturazioni di schemi ER sulla base di dati quantitativi.

Autonomia di giudizio

Alla fine di questo insegnamento lo studente saprà:

- progettare una piccola base di dati analizzandone in autonomia i requisiti e traducendoli in schema concettuale, logico e fisico, motivare e argomentare le scelte progettuali di una piccola base di dati.

Abilità comunicative

Al termine dell'insegnamento lo studente dovrà essere in grado di:

- esporre, in forma scritta e orale, una relazione tecnica che spieghi e motivi le scelte progettuali nel caso di studio proposto.

4. Modalità di verifica dell'apprendimento:

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

L'esame consiste in una prova scritta e in una discussione orale del progetto di laboratorio.

Prova scritta

La prova prevede da 6 a 8 domande in tutto, al fine di verificare l'apprendimento dei concetti di base. Di queste, due o tre domande riguardano argomenti di laboratorio (SQL), le restanti domande riguardano argomenti ed esercizi della parte di teoria. La valutazione è in trentesimi. Il superamento dello scritto permette di accedere a una delle prove orali previste per l'appello in cui lo scritto è stato superato. Nel caso

in cui questa seconda prova non venga superata, lo scritto dovrà essere ripetuto. Lo studente ha diritto al più a tre correzioni dello scritto in un anno accademico.

Discussione orale del progetto di laboratorio

Viene assegnato un progetto (su argomenti di progettazione) che può essere svolto individualmente o in gruppi formati da 2 o 3 studenti. La discussione della prova di laboratorio è individuale e verte sull'intero progetto. La valutazione è in trentesimi. La parte di laboratorio riguardante l'SQL verrà invece valutata contestualmente allo scritto.

Voto finale

Il voto finale sarà la media pesata dei voti ottenuti nelle due prove (scritta e orale di laboratorio), valutate in trentesimi, essendo comunque necessario il raggiungimento della sufficienza (18/30) in ciascuna prova.

5. Modalità d'insegnamento:

L'insegnamento è strutturato in 40 ore di didattica frontale e 40 ore di attività di laboratorio, suddivise in

lezioni da 2 e 3 ore in base al calendario accademico. Durante l'insegnamento sono proposte agli studenti alcune verifiche dell'apprendimento da svolgersi in aula ed in modo collettivo che propongono esercizi volti a verificare la comprensione degli argomenti trattati.

La frequenza è facoltativa, consigliata, e la prova finale sarà uguale per frequentanti e non.

6. Attività di supporto:

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

7. Programma:

Fondamenti teorici delle basi di dati relazionali (5 CFU, 40 ore):

- Introduzione ai sistemi informativi e alle basi di dati

Il modello relazionale

L'algebra relazionale, con progetto di

espressioni in algebra Introduzione al calcolo relazionale

⋮ Introduzione alle transazioni

Ridondanze e anomalie, dipendenze funzionali e teoria della normalizzazione

Introduzione all'architettura dei sistemi di gestione delle basi di dati: memorizzazione dei dati, indipendenza dei dati, affidabilità, problemi di concorrenza

Laboratorio di Progettazione ed SQL (4 CFU, 40 ore):

- Laboratorio di Progettazione
  1. Metodologie di progetto
  2. Il modello Entity Relationship
  3. La progettazione concettuale
  4. La progettazione logica
  5. Cenni di progettazione fisica
- Laboratorio di SQL con Oracle/PostgreSQL
  1. Il linguaggio SQL e gli standard
  2. Definizione dei dati in SQL (DDL)
  3. Modifica dei dati in SQL (DML)
  4. Interrogazioni in SQL (DML/DQL) e progettazione delle espressioni in SQL

8. Testi consigliati e bibliografia:

Testo di riferimento: Atzeni, Ceri, Fraternali, Paraboschi, Torlone, "Basi di dati", McGraw-Hill, Quinta edizione, 2018.

Materiali aggiuntivi sono forniti dai docenti.

Insegnamento	INF0090 - Calcolabilit? e Complessit?

Insegnamento (inglese):	**Computability and Complexity**
CFU:	**6**
Settore:	**INF/01 - INFORMATICA**
Periodo didattico:	**1**
Tipologia di Attività Formativa:	**B - caratterizzante** **D - libera**
Docenti:	**Stefano BERARDI (Titolare)**

1. Prerequisiti e Propedeuticità:

Competenze attese in ingresso

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

Eventuali corsi propedeutici

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

2. Obiettivi formativi:

Che cos'e' un algoritmo? Quali problemi si possono risolvere con un algoritmo? E in quali casi un algoritmo richiede risorse inaccessibili nella pratica? Il corso affronta questi problemi, trattando anzitutto la teoria della computabilita' sia dal punto di vista matematico - macchine di Turing, funzioni ricorsive - che da prospettive legate ai linguaggi di programmazione, come quella dei programmi while. Si discutono poi i vari possibili criteri di misura delle risorse disponibili (tempo, memoria) e le classi di complessita'. Finiamo presentando il problema P = NP .

3. Risultati dell'apprendimento attesi:

Conoscenza delle definizione del concetto di calcolo, una maggiore consapevolezza delle limitazioni

intrinseche all'uso delle macchine ed un'idea delle strategie per sopperire a tali limitazioni.

4. Modalità di verifica dell'apprendimento:

Esame orale.

5. Modalità d'insegnamento:

Le lezioni si svolgono in aula, ed utilizzano la piattaforma Moodle sia per inviare messaggi agli studenti che per le verifiche dell'apprendimento durante il corso.

6. Attività di supporto:

Vedi il sito moodle del corso

7. Programma:

Teoria della computabilita'

- Le Macchine di Turing
- Problemi non risolubili

Funzioni ricorsive

Calcolabilita' e Linguaggi di Programmazione

Teoria della complessita'

Misure : e classi di Complessita'

Classi : di Complessita' Temporale

Classi di Complessita' Spaziale

Le classi P ed NP

Problemi NP completi

8. Testi consigliati e bibliografia:

L'**unico libro di testo del corso** e' il seguente: ***M. Sipser:"Introduction to the theory of Computation", Course Technology Ptr., 3^edition, 2012.*** Le sezioni effettivamente svolte verranno indicate alla fine dell'anno.

(Per chi desiderasse rivedere in italiano i concetti del testo inglese, segnaliamo il testo: C. Toffalori et alii, Teoria della calcolabilita' e della complessita', McGraw-Hill 2005. Ribadiamo che, tuttavia, questo testo non viene utilizzato nel corso.)

Insegnamento	**MFN0588 - Calcolo Matriciale e Ricerca Operativa**

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

1. Prerequisiti e Propedeuticità:

Competenze attese in ingresso

Nozioni di base di algebra e insiemistica comuni nei programmi di matematica della scuola superiore.

Eventuali corsi propedeutici

Nessuno.

2. Obiettivi formativi:

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

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

3. Risultati dell'apprendimento attesi:

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

Lo studente deve acquisire la capacità costruire modelli di programmazione lineare sia a variabili continue che a

variabili intere partendo dall'enunciato di un problema reale, e la conoscenza dei meccanismi di base e la teoria relativa agli algoritmi che operano su tali modelli.

4. Modalità di verifica dell'apprendimento:

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

Regole da seguire durante la partecipazione ad una prova scritta:

- E' vietata la comunicazione, sotto ogni forma, sia tra persone in aula che fuori dall'aula. La violazione di questa regola comporta l'annullamento della prova.
- E' permesso portare come materiale il formulario del corso.
- E' vietato portare ed utilizzare gli appunti.
- E' permesso portare la calcolatrice.
- Autovalutazione in 3 passi:

1. Le soluzioni dell'esame saranno rese disponibili on-line subito dopo la prova scritta.
2. Lo studente che ha consegnato l'esame ha 24 ore di tempo per

decidere di ritirare la propria prova.

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

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

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

Regole per il calcolo del voto di un esame:

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

5. Modalità d'insegnamento:

Lucidi proiettati in aula e tradizionali spiegazioni alla lavagna.

6. Attività di supporto:

Verranno forniti appunti ed eserciziari, disponibili sulla pagina I-learn del corso.

7. Programma:

1. Vettori e matrici. Operazioni fondamentali. Insiemi convessi, poliedri.
2. Soluzione di un sistema di equazioni lineari.
3. Combinazioni lineari, indipendenza lineare.
4. Programmazione lineare.
   - Modellazione.
   - Struttura della regione ammissibile. Metodo grafico di soluzione.
   - Soluzioni di base. Algoritmo del simplesso.

8. Testi consigliati e bibliografia:

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

Insegnamento	**MFN0604 - Economia e Gestione dell'Impresa e Diritto**

Insegnamento (inglese):	**Management, business administration and computer law**
CFU:	**9**
Settore:	**IUS/02 - DIRITTO PRIVATO COMPARATO** **SECS-P/08 - ECONOMIA E GESTIONE DELLE IMPRESE**
Periodo didattico:	**2**
Tipologia di Attività Formativa:	**C - affine e integrativa**
Docenti:	**Fabio MONTALCINI (Professore a Contratto)** **Marco PIRONTI (Titolare)** **Camillo SACCHETTO (Professore a Contratto)**

1. Prerequisiti e Propedeuticità:

Competenze attese in ingresso

Il corso non presuppone conoscenze iniziali specifiche

Eventuali corsi propedeutici

NESSUNO

2. Obiettivi formativi:

Modulo di Economia

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

Verranno studiate le relazioni delle aziende all'interno della catena del valore (Clieti /fornitori) e i loro settori di riferimento. I framework teorici saranno poi applicati per l'analisi di aziende, settori e mercati innovativi al fine di valutare come vengono colte le opportunità che l'innovazione dà alle aziende e ipotizzare sviluppi futuri di modelli. durante il corso gli

studenti si confronteranno con casi aziendali reali e con imprenditori. gli studeneti del corso potranno inoltre partecipare al progetto silicon valley study tour e passare una settimana in Silicon Valley dove conoscere le aziende più innovative dell'IT.

Gli studenti hanno inoltre l'opportunità di partecipare: www.ggi-academy.it/ *Modulo di Diritto*
Agli studenti saranno forniti le nozioni, i metodi, i modelli cognitivi e gli strumenti tecnici necessari alla comprensione dei concetti fondamentali del Diritto dell'Informatica, settore della scienza forense che realizza maggiormente il dialogo teorico e la sintesi operativa tra il diritto (in particolar modo nei suoi profili pratici e processuali) e l'informatica. Durante il Corso si analizzeranno i principali temi della materia anche alla luce dei piu' significativi casi nazionali e internazionali.

3. Risultati dell'apprendimento attesi:

Modulo di Economia

CREAZIONE E ANALISI DEI MODELLI DI BUSINESS

CREAZIONE DELLE STRATEGIE AZIENDALI ANALISI DEI COMPETITOR, DEL SETTORE, DEL MERCATO.

SOFT SKILL: LAVORO IN TEAM CAPACITA' DI PRESENTARE PROBLEM SOLVING *Modulo di Diritto*
Lo studente apprenderà il know-how normativo, concettuale ed operativo (sostanziale e processuale) per l'analisi e la verifica delle principali fattispecie giuridiche inerenti le nuove tecnologie dell'informazione attraverso lo studio delle sentenze, dei casi pratici e delle regole concrete fondamentali relative ai principali procedimenti – nazionali ed internazionali – nei quali sono coinvolti l'Informatica ed il Diritto.

4. Modalità di verifica dell'apprendimento:

Modulo di Economia

esame scritto sul materiale delle lezioni

Modulo di Diritto

L'apprendimento degli argomenti trattati nel corso sarà valutato mediante una verifica in cui saranno accertate le conoscenze degli aspetti informatico giuridici affrontati.

5. Modalità d'insegnamento:

frontali

6. Attività di supporto:

7. Programma:

Modulo di Economia

business model struttura dell'organizzazione struttura dei processi struttura delle funzioni strategie analisi di settore analisi di mercato analisi unita' organizzativa e processi aziendali *Modulo di Diritto*

Responsabilità Internet Service Provider; Computer Crimes e Prove Digitali; Documento informatico, Firme elettroniche e Posta Elettronica Certificata (PEC); Proprietà intellettuale in ambiente web; Privacy: Principi generali e suoi ambiti peculiari; E-commerce e web contracts.

8. Testi consigliati e bibliografia:

Modulo di Economia

Creare modelli di business. Un manuale pratico ed efficace per ispirare chi deve creare o innovare un modello di business Autore Osterwalder Alexander; Pigneur Yves altri riferimenti bibliografici da definire (comunicati via moodle)

Modulo di Diritto

I materiali didattici saranno suggeriti e forniti dai docenti durante il corso.

Insegnamento	**MFN0617 - Economia e Gestione dell'Innovazione**

Insegnamento (inglese):	**Managing Innovation**
CFU:	**6**
Settore:	**SECS-P/08 - ECONOMIA E GESTIONE DELLE IMPRESE**
Periodo didattico:	**1**
Tipologia di Attività Formativa:	**D - libera**

Docenti: **Marco PIRONTI (Titolare)**

1. Prerequisiti e Propedeuticità:

Competenze attese in ingresso

Conoscenza dei concetti base di Economia e Gestione delle Imprese: analisi di settore analisi di mercato/consumatore modello di business analisi delle organizzazioni strategie delle aziende

Eventuali corsi propedeutici

Corso di Economia e Gestione delle Imprese

2. Obiettivi formativi:

Questo corso ha l'obiettivo di fornire gli skill, tools e approcci per la creazione di un' innovazione sostenibile e di successo nell'era digitale. il corso è multidisciplinare e costruito sulle ultime ricerche legate ai framework di gestione dell'innovazione, sulle metodologie lean per la creazione dei prototipi, sull'additive manufacturing e sull'approccio del design

thinking. Insieme, esploreremo come identificare le opportunità dell'economia digitale e creare un'innovazione che va ben oltre il semplice ricopiare o reinventare un prodotto o servizio risolvendo problemi reali della nostra società quali creazione di lavoro, diminuzione di side effect eliminazione di colli di bottiglia all'interno di prodotti, modelli e settori. identificheremo inoltre framework informatici per facilitare l'identificazione,lo sviluppo e la creazione di innovazione in modo economico e sostenibile. l'obiettivo del corso non sarà solo acquistare gli strumenti teorici per creare un innovazione ma anche riuscire ad applicarli per arrivare a creare un'innovazione di prodotto o servizio all'interno di un mercato di riferimento Gli studenti del corso parteciperanno alla European innovation Accademy

3. Risultati dell'apprendimento attesi:

capacità di gestire un processo di creazione dell'innovazione capacità di utilizzare diversi approcci quali: 1)job to be done per il riconoscimento dell'innovazione e dei clienti 2)lean methodology: per la creazione del MINIMUM VIABLE PRODUCT e i testing

dei prototipy 3)design thinking: per la comunicazione del prodotto e strumenti

4. Modalità di verifica dell'apprendimento:

le modalità d'esame sono differenti a seconda che lo studente sia

1) modalità di esame per i frequentanti 70% valutazione attraverso una prova scritta 30% progetto di gruppo e partecipazione attiva in aula 2)non frequentanti: 70% valutazione attraverso una prova scritta 30%progetto individuale

5. Modalità d'insegnamento:

Le lezioni si basano su lezioni teoriche e laboratori pratici

6. Attività di supporto:

le slide del corso verranno inserite sul sito del corso

7. Programma:

---il consumatore ---l'individuazione dell'opportunità ---

la value proposition ---creazione del prototipo --Test Lean ---sperimentazione ---design/comunicazione e visualizzazione

8. Testi consigliati e bibliografia:

riferimenti bibliografici da definire (comunicati via moodle)

Insegnamento	**MFN0600 - Elementi di Probabilità e Statistica**

Insegnamento (inglese):	**Foundations of Probability and Statistics**
CFU:	**6**
Settore:	**MAT/06 - PROBABILITA\' E STATISTICA MATEMATICA**
Periodo didattico:	**1**
Tipologia di Attività Formativa:	**C - affine e integrativa**
Docenti:	**Roberta SIROVICH (Titolare)**

1. Prerequisiti e Propedeuticità:

Competenze attese in ingresso

Calcolo di base.

2. Obiettivi formativi:

L'insegnamento si inserisce nell'area tematica della matematica, i
statistica. Si propone il duplice obiettivo di fornire allo studente sia la conoscenza e la capacità di comprensione dei fenomeni di natura aleatoria sia gli strumenti metodologici e analitici correlati, che siano di supporto per i corsi successivi ma anche di valore intrinseco. L'introduzione ai concetti di rischio e di probabilità fornisce gli strumenti analitici e modellistici per la trattazione di eventi casuali. L'introduzione alla statistica fornisce gli strumenti metodologici per trattare con le quantità aleatorie rilevabili.

3. Risultati dell'apprendimento attesi:

Lo studente avrà familiarità con il calcolo della probabilità di eventi, il concetto di variabile
aleatoria e con le principali distribuzioni di probabilità. Avrà acquisito i concetti base della statistica descrittiva e inferenziale. Sarà inoltre in grado di svolgere le principali analisi dati considerate standard. Lo studente a tale scopo apprenderà l'utilizzo del software R.

4. Modalità di verifica dell'apprendimento:

La verifica della preparazione dello studente avviene con un esame scritto. L'esame è svolto in laboratorio sotto forma di un quiz Moodle.

5. Modalità d'insegnamento:

Lezioni ed esercitazioni frontali in aula.

6. Attività di supporto:

7. Programma:

Il programma di lavoro è articolato in due parti, la probabilità e la statistica.

Probabilità. Spazio campionario e probabilità, insiemi, modelli probabilistici, probabilità condizionata, teorema delle probabilità totali, formula di Bayes, indipendenza. Variabili aleatorie discrete, funzione di densità discreta, funzioni di variabili aleatorie, attesa, varianza. Densità

discreta congiunta di variabili multidimensionali, condizionamento e indipendenza. Variabili aleatorie continue, funzione di densità, funzione di distribuzione cumulata. Densità di probabilità variabili aleatorie multidimensionali. Covarianza e correlazione. Legge dei grandi numeri e teorema del limite centrale.

Statistica. Introduzione a R. Dati univariati, tipi di dati, indici riassuntivi di posizione, di variabilità e di forma. Rappresentazioni grafiche, istogrammi e box plot. Dati invariati, scatterplot. Confronti qualitativi, dati appaiati. Dati invariati categoriali e tabelle. Intervalli di confidenza (proporzioni, medie, varianze, differenze di medie). Test di ipotesi (proporzioni, medie).

8. Testi consigliati e bibliografia:

DP Bertsekas and JN Tsitsiklis. *Introduction to probability*. 2nd Edition. Athena Scientific, 2008.

J Verzani. *Using R for Introductory Statistics*. 2nd Edition. CRC Press, 2014.

SM Ross. *Introduction to Probability and Statistics for Engineers and Scientists*. 5th Edition. Academic Press, 2014.

Insegnamento	MFN0598 - Fisica

Insegnamento (inglese):	Physics
CFU:	6
Settore:	FIS/01 - FISICA SPERIMENTALE
Periodo didattico:	2
Tipologia di Attività Formativa:	C - affine e integrativa
Docenti:	Martino GAGLIARDI (Titolare) Ernesto MIGLIORE (Titolare) Igor PESANDO (Titolare)

1. Prerequisiti e Propedeuticità:

Competenze attese in ingresso

Competenze di base di trigonometria, di calcolo vettoriale e di analisi matematica.

Eventuali corsi propedeutici

Corsi di Matematica del I anno.
Nella prima settimana di lezione si svolgera` un ciclo di lezioni (8h) di richiamo sulla *Meccanica del punto materiale* (argomenti trattati: richiami di cinematica; leggi di Newton; lavoro ed energia; forze conservative).

2. Obiettivi formativi:

L'insegnamento si propone di:

1. introdurre alla conoscenza della basi indispensabili di meccanica, delle principali proprietà del campo elettrico e del campo magnetico, con cenni al comportamento della materia soggetta a tali campi;
2. introdurre alla conoscenza del comportamento degli elementi di un circuito in corrente continua ed in corrente alternata;
3. introdurre alla conoscenza dei principi fisici alla base del funzionamento delle porte logiche.

3. Risultati dell'apprendimento attesi:

Lo studente deve avere acquisito la capacità di

esprimere i concetti con un formalismo matematico sufficientemente rigoroso e di collegare fra di loro in modo armonico i principi fondamentali della fisica. Deve inoltre essere in grado di fornire una risposta quantitativa ai problemi assegnati.

4. Modalità di verifica dell'apprendimento:

L'esame di Fisica è costituito dalla sola prova scritta, comune per il corso A e per il corso B. Il formato dell'esame (2h a disposizione) è il seguente:

1. 1 esercizio sull'algebra dei vettori
2. 1 esercizio su campo elettrico oppure su campo magnetico + 1 esercizio su circuiti
3. 1 domanda aperta da svolgere (max 1 facciata) tra 2 estratte, al momento dello scritto, da questa lista

Per superare l'esame occorre soddisfare le seguenti condizioni (operazione logica AND fra le condizioni):

avere ⋮ fatto tutto giusto l'esercizio sui vettori
avere svolto >50% complessivo dei due esercizi su
campi e circuiti avere risposto in modo esaustivo
alla domanda aperta

Il primo esercizio e` di sbarramento e **non** conta per il voto finale. La restante parte di esercizi e la parte di teoria contribuiscono in **egual** misura alla determinazione del voto finale.
Il >50% complessivo dei due esercizi puo' esser ottenuto per esempio con 10%+45% o 30%+30% dove il primo valore percentuale si riferisce all'esercizio sull'elettrmagnetismo ed il secondo all'esercizio sui circuiti.

All'esame e` possibile utilizzare la calcolatrice.
Non e` consentito l'uso di libri o appunti ma solo di un formulario distribuito dai docenti in sede di esame. Gli scritti assegnati negli ultimi anni e le relative soluzioni sono reperibili al seguente link

5. Modalità d'insegnamento:

L'insegnamento si svolge attraverso lezioni frontali ed esercitazioni svolte dagli stessi docenti. Nella prima settimana di lezione e` previsto un ciclo di lezioni (8h)

di richiamo sulla *Meccanica del punto materiale*.

6. Attività di supporto:

Supporto tramite web

L'insegnamento e` presente sulla piattaforma di e-learning. Il forum dell'insegnamento e` usato abitualmente per le comunicazioni con gli studenti e per la distribuzione del materiale didattico. Il materiale relativo alle esercitazioni e` reperibile al link: http://personalpages.to.infn.it/~gagliard/

Tutoraggio in aula

Vengono fornite ore di tutoraggio (2 ore/settimana) in cui si ha la possibilità di svolgere gli esercizi sotto la guida di un tutore. Gli esercizi proposti sono scelti in stretto rapporto del docente dell'insegnamento con il tutore.

TUTORATO a.a. 2018-2019

Tutorato, II semestre lunedi` 14-16 (eventuali variazioni sono comunicate tramite moodle).

7. Programma:

Il campo elettrostatico. Legge di Gauss. 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 e di Kirchhoff. Circuiti RC.

Fisica dei dispositivi elettronici. Conduttori, isolanti e semiconduttori. Giunzione pn e diodo. Dispositivi MOS e porte logiche (NOT, NOR, NAND).

Il campo magnetico indipendente dal tempo. Magneti. 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. Legge di Ampere. Legge 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.

Il programma dettagliato relativo all'insegnamento sara` pubblicato dai docenti al termine delle lezioni.

E` possibile consultare anche il registro delle lezioni dell'A.A. 2017-2018:

- programma dettagliato corso A (A.A. 2017/2018) pdf programma dettagliato corso B (A.A. 2012/2013) pdf

8. Testi consigliati e bibliografia:

Testi consigliati (tra cui scegliere):

1. WOLFSON, Fisica, vol.2 - Elettromagnetismo, ottica e fisica moderna, Pearson
2. RESNICK/HALLIDAY/KRANE, Fisica, vol. 2, Casa Editrice Ambrosiana

3. SERWAY/JEWETT Fisica. Per scienze ed ingegneria, vol. 2, EdiSes

Insegnamento	MFN1353 - Interazione Uomo Macchina

Insegnamento (inglese):	**Human-computer Interaction**
CFU:	**6**
Settore:	
Periodo didattico:	**1**
Tipologia di Attività Formativa:	**D - libera**
Docenti:	**Viviana PATTI (Titolare)** **Marino SEGNAN (Titolare)**

1. Prerequisiti e Propedeuticità:

Competenze attese in ingresso

Si richiede una buona conoscenza delle basi di dati (fornita dal corso di Basi di Dati), dell'analisi e della progettazione object-oriented (fornita dai corsi di Programmazione II e di Algoritmi e strutture dati) e dei fondamenti della programmazione distribuita (fornita dal

corso di
Programmazione III). Lo studente deve inoltre avere la capacità di scrivere, compilare e verificare la correttezza di programmi in Java.

Eventuali corsi propedeutici

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

2. Obiettivi formativi:

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

influenzare non solo le prestazioni dell'applicazione, la sua scalabilità e robustezza, ma anche le tipologie di servizio che possono effettivamente essere offerte.

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

Più precisamente, per quanto riguarda l'interazione uomo-macchina, verranno acquisite competenze sia teoriche sia pratiche nel disegno corretto di interazioni, con specifico riferimento alla progettazione user-centered. Per quanto riguarda gli aspetti architetturali e tecnologici, il corso tratterà dal punto di vista sia teorico che pratico la programmazione lato client per device mobili su piattaforma Android e lo sviluppo di interfacce mobili. Per permettere agli studenti di

sperimentare le nozioni apprese durante le ore di teoria in aula il corso include una sostanziale parte di laboratorio. I temi introdotti durante il laboratorio corredano e integrano le conoscenze derivanti dalla parte teorica (knowledge and understanding) e permettono agli studenti di familiarizzare con le metodologie e tecnologie introdotte, anche investigando soluzioni alternative (applying knowledge and understanding).

Inoltre durante le ore di laboratorio è previsto lo sviluppo di un'applicazione realistica con interfaccia mobile ed una introduzione al linguaggio Python. La preparazione e la discussione del progetto di laboratorio sono inoltre volte a stimolare le capacità di organizzare il lavoro in piccoli gruppi (max 4 studenti), e poi di illustrare verbalmente le soluzioni adottate (communication skills).

3. Risultati dell'apprendimento attesi:

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

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

4. Modalità di verifica dell'apprendimento:

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

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

VERIFICA DI LABORATORIO: prevede la discussione del progetto di laboratorio svolto durante il corso. La discussione deve essere effettuata preferibilmente in

unica soluzione, con tutti i membri del gruppo di laboratorio presenti. Il voto di laboratorio è un numero compreso tra 0 e 15, si considerano sufficienti i voti >=9. CALCOLO DEL VOTO FINALE DI ESAME: Sia X il voto del test scritto; sia Y il voto di laboratorio. Il voto Fin finale dell'esame si ottiene come segue: Fin = (X+Y)

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

5. Modalità d'insegnamento:

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

La parte di laboratorio consiste di 30 ore ed è

focalizzata sulla programmazione client-side in ambiente Android utilizzando le API Java del sistema operativo. Verra' inoltre sviluppata un' applicazione scritta in linguaggio Python . Le lezioni si svolgono in maniera interattiva e sono corredate da vari esercizi miranti a fornire esempi pratici.

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

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

6. Attività di supporto:

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

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

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

7. Programma:

- Parte Ia - Human-computer interaction (HCI)
  - Human-computer interaction (HCI): Definizioni e contesto, evoluzione di HCI, nuove direzioni.

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

    Disegno di interazioni: user-centered design, requisiti funzionali e di usabilità (raccolta, analisi, presentazione), prototipazione, linee

guida (con gestione degli errori ed assistenza agli utenti), elementi di tipografia elettronica, di layout e gestione del colore.

- Tecniche di valutazione: valutazione senza utenti (quantitativa e qualitativa), valutazione con utenti, problemi, presentazione dei risultati. Disegno inclusivo: accessibilità, disegno per utenti di differenti gruppi di età (bambini, anziani), internazionalizzazione.

- Parte Ib - Programmazione di device mobili.
  - Introduzione alla programmazione per mobile.
  - La piattaforma Android e sua architettura. Processi e applicazioni in Android.
  - Il linguaggio Python:differenze rispetto a Java
  - Progettazione di una interfaccia utente in maniera programmatica e dichiarativa.
  - Sviluppare con Python o Java? Confronto tra gli ambienti ed esempi
  - Esempio di sviluppo del lato client di una semplice app per Android.

8. Testi consigliati e bibliografia:

Libri di Testo:

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

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

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

Python. http://pythonspot.com/beginner/

Altri testi (per consultazione):

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

Insegnamento	**MFN0608 - Interazione Uomo Macchin Tecnologie Web**

Insegnamento (inglese):	**Human-computer Interaction and Web Technologies**
CFU:	**12**
Settore:	**INF/01 - INFORMATICA**
Periodo didattico:	**1**
Tipologia di Attività Formativa:	**B - caratterizzante**

Docenti:

Liliana ARDISSONO (Titolare)
Viviana PATTI (Titolare)
Marino SEGNAN (Titolare)

---

1. Prerequisiti e Propedeuticità:

Competenze attese in ingresso

Si richiede una buona conoscenza delle basi di dati (fornita dall'insegnamento Basi di Dati), dei sistemi operativi (fornita dall'insegnamento Sistemi Operativi), dell'analisi e della progettazione object-oriented (fornita dagli insegnamenti Programmazione II, e Algoritmi e strutture dati) e dei fondamenti della programmazione distribuita (fornita dall'insegnamento Programmazione III). Gli studenti e le studentesse devono inoltre avere la capacità di scrivere, compilare e verificare la correttezza di programmi in Java.

Eventuali corsi propedeutici

Basi di Dati, Sistemi Operativi, Programmazione II, Algoritmi e strutture dati, Programmazione III

2. Obiettivi formativi:

Questo insegnamento si colloca nell'area di approfondimento del percorso formativo del Corso di Laurea in Informatica e ha come obiettivi formativi specifici l'insegnamento dei concetti fondamentali dell'Interazione Uomo-Macchina e delle metodologie per le applicazioni web e mobile.

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

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

Più precisamente, per quanto riguarda l'interazione uomo-macchina, verranno acquisite competenze sia teoriche sia pratiche nel disegno corretto di interazioni, con specifico riferimento alla progettazione user-centered. Per quanto riguarda gli aspetti architetturali e tecnologici, nell'insegnamento verranno acquisite competenze teoriche e pratiche su: (i) lo sviluppo di pagine web statiche e dinamiche responsive; (ii) la programmazione lato server; (iii) la programmazione lato client per device mobili su piattaforma Android e lo sviluppo di interfacce mobili per applicazioni web. Inoltre, l'insegnamento fornira' conoscenza relativa alla rappresentazione e

interpretazione di dati con XML, data la sua importanza per l'interoperabilità tra applicazioni.

In questo insegnamento, le capacita' di applicare le conoscenze acquisite vengono sviluppate nell'ambito di una sostanziale parte di laboratorio che permette di sperimentare le nozioni apprese durante le ore di teoria in aula, organizzando il lavoro in piccoli gruppi. I temi introdotti durante il laboratorio corredano e integrano le conoscenze derivanti dalla parte teorica (knowledge and understanding) e permettono agli studenti e alle studentesse di familiarizzare con le metodologie e tecnologie introdotte, anche investigando soluzioni alternative (applying knowledge and understanding). Inoltre durante le ore di laboratorio è previsto lo sviluppo di un'applicazione realistica con interfaccia sia web che mobile, che va oltre la complessita' del tipico esercizio svolto durante le ore di lezione. Lo svolgimento del progetto di laboratorio permette di allenare l'autonomia di giuzio nella scelta delle implentazioni da realizzare, analizzando in forma critica le informazioni acquisite nella parte teorica

dell'insegnamento. La preparazione del progetto di laboratorio e' anche volta a stimolare la capacità comunicativa degli studenti e delle studentesse che sono invitate/i a illustrare verbalmente le soluzioni adottate (communication skills).

3. Risultati dell'apprendimento attesi:

Gli studenti e le studentesse acquisiranno conoscenza su:

- progettazione user-centered, la prototipazione, la
- valutazione dell'usabilità e il disegno inclusivo per
- categorie speciali di utenti.

concetti rilevanti per analizzare l'interazione: affordance, modelli concettuali, metafore, feedback, etc. architetture di riferimento per lo sviluppo di applicazioni web e mobile modelli più usati per gestire applicazioni interattive di media complessita` tecnologie attualmente utilizzate per l'implementazione delle applicazioni (back-end e interfacce utente web/mobili).

Gli studenti e le studentesse acquisiranno anche la capacita` di sviluppare

- pagine dinamiche PHP
- applicazioni Web a 3 livelli in ambiente Java, utilizzando HTML5-JavaScript, CSS, JSP e Java Servlet, e framework per applicazioni web MVC applicazioni mobili.

Per quanto riguarda questo ultimo punto, essi/esse saranno in grado di realizzare una applicazione all'interno del sistema Android che interagisca correttamente coi servizi del sistema operativo e le altre applicazioni residenti. Saranno inoltre in grado di saper scrivere un programma usando i costrutti piu' espressivi del linguaggio Python.

4. Modalità di verifica dell'apprendimento:

L'esame è composto da un TEST SCRITTO e da una VERIFICA DI LABORATORIO. Il primo e' volto a verificare che gli studenti e le studentesse abbiano acquisito la conoscenza fornita dall'insegnamento in termini di sapere. Il secondo e' volto a verificare le competenze acquisite (saper fare), nonche' le abilita' comunicative e l'autonomia di giudizio. Le due prove possono essere sostenute in

qualsiasi ordine.

TEST SCRITTO: Prova scritta che include esercizi e domande teoriche sul programma dell'insegnamento. Viene valutata da un minimo di 0 ad un massimo di 30 e si considerano sufficienti i voti >=18. Come da regolamento di Ateneo, ogni studente/studentessa può sostenere un numero massimo di tre prove scritte durante l'Anno Accademico (cioè, consegnare il proprio elaborato tre volte). Il voto ottenuto durante un test scritto decade se si partecipa ad un altro test scritto e si consegna l'elaborato.

VERIFICA DI LABORATORIO: consiste di (1) la discussione di gruppo di un progetto Web svolto durante il corso, che include una applicazione java con interfaccia Web e mobile Android, e la realizzazione di un programma scritto in linguaggio Python. La discussione deve essere effettuata preferibilmente in unica soluzione, con tutti i membri del gruppo di laboratorio presenti. Il voto di laboratorio è un numero intero compreso tra 0 e 30, si considerano sufficienti i voti >=18.

CALCOLO DEL VOTO FINALE DI ESAME: Sia X il voto del test scritto; sia Y il voto di laboratorio.

Il voto Fin finale dell'esame si ottiene come segue: Fin = (X+Y)/2
Note: i voti acquisiti durante la prova di laboratorio, o durante il test scritto, rimangono validi fino al termine della terza sessione d'esame (quella che precede l'inizio del nuovo corso). Quando si superano entrambe le prove, è necessario registrare il voto finale entro i limiti imposti dal Regolamento di Ateneo.

5. Modalità d'insegnamento:

L'insegnamento è diviso in una parte di teoria e una di laboratorio. Per la parte di teoria sono previste 48 ore di lezione frontali che seguono il programma dell'insegnamento, integrate da casi di studio e da esercitazioni volte ad illustrare l'applicazione pratica dei concetti appena studiati.

La parte di laboratorio è articolata in due moduli da 30 ore ciascuno. Il primo modulo ha per oggetto l'insegnamento di tecnologie per lo sviluppo di pagine web statiche e dinamiche, e di applicazioni server-side MVC con accesso a basi di dati relazionali. Il secondo

modulo è suddiviso in due parti: programmazione client-side in ambiente Android utilizzando il linguaggio Java; scrittura di un programma in Python. Le esercitazioni in laboratorio sono svolte a calcolatore, utilizzando ambienti di sviluppo professionali, come IntelliJ IDEA e/o NetBeans IDE.

Le sperimentazioni che vengono effettuate durante le ore di laboratorio, strutturate come sequenze di esercizi specifici, sono fondamentali per aiutare gli studenti e le studentesse a comprendere e assimilare i contenuti teorici spiegati a lezione in quanto permettono di mettere in pratica i concetti e le metodologie illustrate su esempi concreti. Inoltre lo sviluppo del progetto di laboratorio, che si effettua in gruppo (max 3 persone per gruppo), permette di consolidare le conoscenze teoriche e il lavoro in team in un caso realistico di media complessità.

Si consiglia caldamente la frequenza costante alle lezioni di teoria e di laboratorio. È inoltre fondamentale iscriversi al corso online su I-Learn, all'interno del quale e' messo a disposizione materiale didattico di supporto.

6. Attività di supporto:

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

Il materiale didattico di supporto (lucidi delle lezioni, link a documentazione, esempi di testo di esami ed altro) è disponibile presso il supporto on-line ai corsi I-learn (vd. le pagine dei corsi (i) Interazione Uomo Macchina e (ii) Interazione Uomo Macchina e Tecnologie Web - parte tecnologica).

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

7. Programma:

PARTE 1A: HUMAN-COMPUTER INTERACTION (HCI)

- Human-computer interaction (HCI): Definizioni e contesto, evoluzione di HCI, nuove direzioni.

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

Disegno di interazioni: user-centered design, requisiti funzionali e di usabilità (raccolta, analisi, presentazione), prototipazione, linee guida (con gestione degli errori ed assistenza agli utenti), elementi di tipografia elettronica, di layout e gestione del colore.

- Tecniche di valutazione: valutazione senza utenti (quantitativa e qualitativa), valutazione con utenti, problemi, presentazione dei risultati.

Disegno inclusivo: accessibilità, disegno per utenti di differenti gruppi di età (bambini/e, persone anziane), internazionalizzazione.

PARTE 1B - PROGRAMMAZIONE DI DEVICE MOBILI

- Il sistema operativo Android e le sue peculiarita'. Gli elementi costitutivi di una applicazione utente ed il suo ciclo di vita. Le pratiche consolidate e l'ambiente di sviluppo di una applicazione Android.
- Il linguaggio avanzato Python: principali differenze rispetto al linguaggio Java e le sue caratteristiche piu' interessanti.

PARTE 2 - APPLICAZIONI WEB E TECNOLOGIE DI SUPPORTO

- Architetture delle applicazioni Web: Web browser e

Web server.

Il Pattern Model View Controller per le applicazioni Web - Progettazione e sviluppo di applicazioni Web a 3 livelli basate su MVC:

- Il primo livello (client dell'applicazione): HTML5, CSS, scripting lato client (JavaScript e AJAX), Java Server Pages (JSP). Raccolta dati con HTML form.
- Il secondo livello (logica applicativa): Servlet Java.
- Il terzo livello (livello dei dati): accesso a database relazionali con Java Database Connectivity (JDBC), rappresentazione e gestione di informazioni con XML (XML Schema, XPath, XML Parsers).
- Framework le applicazioni web MVC.

Introduzione al linguaggio PHP per lo sviluppo di pagine web dinamiche con accesso a DB relazionale.

8. Testi consigliati e bibliografia:

PARTE 1A:

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

PARTE 1B:

- Android: http://www.html.it/guide/guida-android/
- Python: http://www.python.it/doc/newbie/

PARTE 2:

- Programmazione Web lato Server, di V. Della Mea, L. Di Gaspero, I. Scagnetto, Edizione 2, Apogeo, 2011

ALTRI RIFERIMENTI (per consultazione):

- La caffettiera del masochista, di D. Norman, Apogeo
- Programmazione Java - tecniche avanzate, di Deitel e Deitel. Ed. Pearson - Prentice Hall Principi di Web design, di Joe Sklar, Apogeo

Insegnamento	**MFN0590 - Lingua Inglese I**

Insegnamento	**English I**

(inglese):	
CFU:	**3**
Settore:	**L-LIN/12 - LINGUA E TRADUZIONE - LINGUA INGLESE**
Periodo didattico:	**1** **2**
Tipologia di Attività Formativa:	**E - prova finale e lingua straniera**
Docenti:	**Enrico BINI (Titolare)** **Viviana BONO (Titolare)** **Francesca CORDERO (Titolare)** **Jeanne Marie GRIFFIN (Esercitatore)** **Viviana PATTI (Titolare)** **Daniele Paolo RADICIONI (Titolare)** **Maddalena ZACCHI (Titolare)**

1. Prerequisiti e Propedeuticità:

Competenze attese in ingresso

Nessuna.

2. Obiettivi formativi:

Esercitazioni di base di Inglese orientate alla comprensione dei testi a alla grammatica di base.

3. Risultati dell'apprendimento attesi:

Al termine dell'insegnamento lo studente deve essere in grado di comprendere e produrre brevi frasi in lingua inglese e leggere semplici testi di argomento scientifico-informatico.

4. Modalità di verifica dell'apprendimento:

Il test finale e` svolto nei laboratori informatici mediante il sistema SET. La prova di Inglese I per informatica e` 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 e` automatica e comunicata immediatamente dal sistema allo studente. In ogni appello si svolgono entrambe le parti, in successione; gli studenti che hanno gia` superato la prima parte in un appello precedente

devono dare solo la seconda.

Attenzione: 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 (http://di.unito.it/aputarm). Il modulo va consegnato on-line sul proprio spazio MyUniTO oppure alla Segreteria Studenti del Polo delle Scienze della Natura (http://di.unito.it/segreteriastudenti), in via S. Croce 6, secondo le scadenze definite dalla Segreteria, di norma a metà giugno di ogni anno. 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.

5. Modalità d'insegnamento:

Le esercitazioni sono svolte in aula dall'esperto linguistico (Dott.ssa Jeanne Griffin).

6. Attività di supporto:

Sono disponibili dispense redatte dall'esperto linguistico.

Le dispense aggiornate sono disponibili anche su I-learn.

Commissione cui eventualmente rivolgersi:

commissione-inglese@di.unito.it

7. Programma:

- Grammatica di base
- Lessico
- Pratica

Informazioni dettagliate su I-learn.

8. Testi consigliati e bibliografia:

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

Insegnamento	**MFN1354 - Linguaggi e Paradigmi di Programmazione**

Insegnamento (inglese):	**Programming languages and paradigms**
CFU:	**6**
Settore:	**INF/01 - INFORMATICA**
Periodo didattico:	**1**
Tipologia di Attività Formativa:	**D - libera**
Docenti:	**Viviana BONO (Titolare)** **Luca PADOVANI (Titolare)**

1. Prerequisiti e Propedeuticità:

Competenze attese in ingresso

Conoscenza delle basi della matematica discreta e della programmazione.

Eventuali corsi propedeutici

Matematica discreta e logica; Programmazione 1 e 2.

2. Obiettivi formativi:

I linguaggi di programmazione moderni supportano molteplici paradigmi di programmazione, quello imperativo e quello orientato agli oggetti in primis ma, recentemente, anche quello funzionale. L'approccio funzionale consente al programmatore di lavorare a un più alto livello di astrazione, favorendo lo sviluppo rapido di prototipi con provabili garanzie di correttezza, la modularità e il riuso del codice. L'insegnamento ha l'obiettivo di approfondire concetti di informatica di base e in particolare fornire una introduzione al paradigma di programmazione funzionale, ai sistemi di tipi per i linguaggi funzionali con i relativi algoritmi di inferenza e alla dimostrazione di correttezza di programmi funzionali. Nella parte avanzata del corso si introdurranno tecniche di programmazione e dimostrazione per strutture dati co-induttive - in particolare gli stream - e si illustrerà l'applicazione delle tecniche apprese a Java il quale, a partire dalla

versione 8, ha incorporato costrutti tipici della programmazione funzionale, in particolare le lambda espressioni.

3. Risultati dell'apprendimento attesi:

Al termine dell'insegnamento lo studente dovrà dimostrare di avere compreso i principi alla base della programmazione funzionale. Dovrà quindi dimostrare di conoscere le principali tecniche di programmazione e dimostrazione utilizzate in tale paradigma e le strutture di dati impiegate (in particolare, le liste e gli stream), utilizzandole per impostare e risolvere problemi di programmazione.

Relativamente alla parte avanzata del corso, lo studente dovrà dimostrare di aver appreso la tecnica di dimostrazione per co-induzione e di saper applicare costrutti e pattern tipici della programmazione funzionale al linguaggio Java.

4. Modalità di verifica dell'apprendimento:

Esame scritto in laboratorio con esercizi su parte teorica e laboratorio, in quest'ultimo caso da svolgere al calcolatore.

5. Modalità d'insegnamento:

Lezioni ed esercitazioni in aula ed esercitazioni in laboratorio, in entrambi i casi con sessioni interattive sia su carta che al computer.

6. Attività di supporto:

Non previste.

7. Programma:

Nel seguente elenco di argomenti non viene fatta distinzione tra argomenti svolti in aula ed argomenti svolti in laboratorio, che sono peraltro strettamente connessi.

- Storia dei linguaggi di programmazione, con particolare riferimento a quelli funzionali;

Calcolo come riscrittura: le basi dell'esecuzione dei programmi funzionali;
Espressioni e loro tipi. Tipi di base;
Progettazione di programmi funzionali. Tecniche di ricorsione;
Liste e funzioni del prim'ordine su liste;
Dimostrazioni di correttezza di programmi funzionali usando l'induzione;
L'idea di astrazione funzionale. Funzioni di ordine superiore e pattern di trasformazione; Alberi e tipi algebrici generali;
Cenni a monadi e programmazione imperativa.

Per la versione da 9 CFU, i seguenti argomenti addizionali:

- Strutture di dati infinite (stream) e dimostrazioni di correttezza usando la co-induzione;
- Programmazione funzionale in Java 8.

8. Testi consigliati e bibliografia:

Simon Thompson, "Haskell: the Craft of Functional Programming" (terza edizione), AddisonWesley, 2011.

Raoul Gabriel Urma, Mario Fusco, Alan Mycroft, "Java 8 In Action: Lambdas Streams And Functional Style Programming", Manning Publications, 2014.

Può anche essere utile la consultazione di:

- Graham Hutton, "Programming in Haskell" (seconda edizione), Cambridge University Press, 2016.
- Richard Bird, "Thinking Functionally with Haskell", Cambridge University Press, 2014.

Insegnamento	**MFN0610 - Linguaggi e Paradigmi di Programmazione**

Insegnamento (inglese):	**Programming languages and paradigms**
CFU:	**9**
Settore:	**INF/01 - INFORMATICA**
Periodo didattico:	**1**
Tipologia di Attività Formativa:	**B - caratterizzante**
Docenti:	**Viviana BONO (Titolare)** **Luca PADOVANI (Titolare)**

1. Prerequisiti e Propedeuticità:

Competenze attese in ingresso

Conoscenza delle basi della matematica discreta e della programmazione.

Eventuali corsi propedeutici

Matematica discreta e logica; Programmazione 1 e 2.

2. Obiettivi formativi:

I linguaggi di programmazione moderni supportano molteplici paradigmi di programmazione, quello imperativo e quello orientato agli oggetti in primis ma, recentemente, anche quello funzionale. L'approccio funzionale consente al programmatore di lavorare a un più alto livello di astrazione, favorendo lo sviluppo rapido di prototipi con provabili garanzie di correttezza, la modularità e il riuso del codice. L'insegnamento ha l'obiettivo di approfondire concetti di informatica di base e in particolare fornire una introduzione al paradigma di programmazione funzionale, ai sistemi di tipi per i linguaggi funzionali con i relativi algoritmi di

inferenza e alla dimostrazione di correttezza di programmi funzionali. Nella parte avanzata del corso si introdurranno tecniche di programmazione e dimostrazione per strutture dati co-induttive - in particolare gli stream - e si illustrerà l'applicazione delle tecniche apprese a Java il quale, a partire dalla versione 8, ha incorporato costrutti tipici della programmazione funzionale, in particolare le lambda espressioni.

3. Risultati dell'apprendimento attesi:

Al termine dell'insegnamento lo studente dovrà dimostrare di avere compreso i principi alla base della programmazione funzionale. Dovrà quindi dimostrare di conoscere le principali tecniche di programmazione e dimostrazione utilizzate in tale paradigma e le strutture di dati impiegate (in particolare, le liste e gli stream), utilizzandole per impostare e risolvere problemi di programmazione.

Relativamente alla parte avanzata del corso, lo studente dovrà dimostrare di aver appreso la tecnica di dimostrazione per co-induzione e di saper applicare

costrutti e pattern tipici della programmazione funzionale al linguaggio Java.

4. Modalità di verifica dell'apprendimento:

Esame scritto in laboratorio con esercizi su parte teorica e laboratorio, in quest'ultimo caso da svolgere al calcolatore. Rispetto alla versione del corso da 6 cfu, la prova di esame comprende esercizi relativi agli argomenti aggiuntivi (stream, programmazione funzionale in Java 8).

5. Modalità d'insegnamento:

Lezioni ed esercitazioni in aula ed esercitazioni in laboratorio, in entrambi i casi con sessioni interattive sia su carta che al computer.

6. Attività di supporto:

Non previste.

7. Programma:

Nel seguente elenco di argomenti non viene fatta distinzione tra argomenti svolti in aula ed argomenti svolti in laboratorio, che sono peraltro strettamente connessi.

- Storia dei linguaggi di programmazione, con particolare riferimento a quelli funzionali;
- Calcolo come riscrittura: le basi dell'esecuzione dei programmi funzionali;

Espressioni e loro tipi. Tipi di base;

Progettazione di programmi funzionali. Tecniche di ricorsione;

Liste e funzioni del prim'ordine su liste;

Dimostrazioni di correttezza di programmi funzionali usando l'induzione;

L'idea di astrazione funzionale. Funzioni di ordine superiore e pattern di trasformazione; Alberi e tipi algebrici generali;

Cenni a monadi e programmazione imperativa.

Per la versione da 9 CFU, i seguenti argomenti addizionali:

- Strutture di dati infinite (stream) e dimostrazioni di correttezza usando la co-induzione;
- Programmazione funzionale in Java 8.

8. Testi consigliati e bibliografia:

Simon Thompson, "Haskell: the Craft of Functional Programming" (terza edizione), AddisonWesley, 2011.

Raoul Gabriel Urma, Mario Fusco, Alan Mycroft, "Java 8 In Action: Lambdas Streams And Functional Style Programming", Manning Publications, 2014.

Può anche essere utile la consultazione di:

Graham Hutton, "Programming in Haskell" (seconda edizione), Cambridge University Press, 2016.

Richard Bird, "Thinking Functionally with Haskell", Cambridge University Press, 2014.

Insegnamento	MFN0603 - Linguaggi Formali e Traduttori

Insegnamento (inglese):	**Formal Languages and Compilers**
CFU:	**9**
Settore:	**INF/01 - INFORMATICA**
Periodo didattico:	**1**
Tipologia di Attività Formativa:	**B - caratterizzante**
Docenti:	**Mario COPPO (Professore a Contratto)** **Luca PADOVANI (Titolare)** **Viviana PATTI (Titolare)** **Jeremy James SPROSTON (Titolare)** **Maddalena ZACCHI (Professore a Contratto)**

1. Prerequisiti e Propedeuticità:

Competenze attese in ingresso

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

linguaggi ad alto livello.

Eventuali corsi propedeutici

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

2. Obiettivi formativi:

Le conoscenze nel campo della descrizione formale dei linguaggi e della traduzione (in particolare della compilazione) sono sempre state considerate fondamentali nel bagaglio culturale di un informatico e non possono essere ignorate dagli addetti al settore. Competenze di questo tipo si trovano nei curricula di orientamento informatico-matematico delle Università di tutto il mondo. I linguaggi di programmazione si sono evoluti presentando nuovi problemi di compilazione che hanno portato allo sviluppo di metodi generali per affrontarli. Buona parte della tecnologia di "front-end" dei compilatori, come grammatiche, espressioni regolari,

parsificatori e traduttori guidati dalla sintassi, trovano anche applicazione in tutti i programmi in cui sia richiesta l'analisi strutturale di un testo o, in generale, di dati in cui si debba individuare una struttura.

L'insegnamento ha l'obiettivo di approfondire i concetti di informatica di base e in particolare dei linguaggi di programmazione fornendo allo studente una visione introduttiva dei problemi connessi alla definizione e alla traduzione dei linguaggi di programmazione, con particolare riferimento al progetto e alla costruzione di compilatori. Le metodologie e le tecniche presentate sono utili in generale come formalismi per definire il comportamento di un sistema o per realizzare componenti che richiedono una traduzione tra rappresentazioni diverse di dati. Scopo del laboratorio è quello di completare la preparazione fornendo competenze pratiche che riguardano l'applicazione dei concetti presentati nelle lezioni di teoria. In particolare l'obiettivo del laboratorio è quello di realizzare un semplice compilatore applicando le tecniche corrette.

3. Risultati dell'apprendimento attesi:

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

4. Modalità di verifica dell'apprendimento:

Ai fini della determinazione del voto finale l'esame dell'insegnamento di Linguaggi Formali e Traduttori è diviso in due parti. 1. Scritto. Lo scritto è formato da: a) domande di teoria su tutto il programma svolto a lezione; b) esercizi che intendono verificare che lo studente sia in grado di applicare quanto appreso.

2. Discussione sull'esercitazione finale di laboratorio consistente nella realizzazione di un interprete/compilatore per un semplice linguaggio definito ad hoc.

Per superare l'esame lo studente deve raggiungere la

sufficienza in entrambe le parti. Per sostenere la prova di laboratorio è necessario aver superato la prova scritta. Entrambe le prove devono essere superate nella stessa sessione d'esame. A richiesta dello studente, e comunque per ottenere la lode, è possibile sostenere una prova orale integrativa.

5. Modalità d'insegnamento:

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

Per la parte di teoria sono previste 48 ore di lezioni frontali. Nel corso delle lezioni vengono anche svolti esercizi esemplificativi. Agli studenti sono resi disponibili le slide eventualmente usate dal docente come supporto alle lezioni.

La parte di laboratorio consiste in un modulo di 30 ore. Agli studenti vengono presentati una serie di esercizi relativi all'applicazione pratica dei concetti trattati nella parte di teoria. Gli esercizi vengono svolti in laboratorio con l'assistenza del docente. I primi esercizi avranno lo scopo di fornire agli studenti l'esperienza necessaria per affrontare l'esercitazione

finale, che richiede la realizzazione di un interprete/compilatore.

6. Attività di supporto:

Le lezioni frontali vengono integrate da 12 ore non creditizzate, con lo scopo di svolgere esercizi e di rispondere alle eventuali richieste da parte degli studenti di approfondimento di argomenti del programma.

Il materiale didattico di supporto (programma dettagliato con riferimento ai capitoli dei libri, slide, esempi di testi d'esame ed altro) sarà reso disponibile sulla piattaforma I-learn.

7. Programma:

Automi a stati finiti ed espressioni regolari, analisi lessicale. Grammatiche e famiglie di linguaggi. Analisi sintattica top-down e bottom-up. Traduzione diretta

dalla sintassi. Generazione del codice intermedio.

Laboratorio: esercizi di programmazione che riguardano, in particolare, l'analisi lessicale (implementazione di automi), l'analisi sintattica (realizzazione di analizzatori a discesa ricorsiva), e la generazione di codice intermedio (sviluppo di un compilatore per un semplice linguaggio).

Il programma dettagliato dell'insegnamento sarà pubblicato sulla piattaforma I-learn

8. Testi consigliati e bibliografia:

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

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

Per la parte di laboratorio, agli studenti viene fornito, oltre ai testi degli esercizi, materiale utile per il loro svolgimento.

Insegnamento	**INF0003 - Logica per l'Informatica**

Insegnamento (inglese):	**Logic for Computer Science**
CFU:	**6**
Settore:	**MAT/01 - LOGICA MATEMATICA**
Periodo didattico:	**2**
Tipologia di Attività Formativa:	**D - libera**
Docenti:	**Luca Luigi PAOLINI (Titolare)**

1. Prerequisiti e Propedeuticità:

Competenze attese in ingresso

- Conoscenze elementari ed informali di logica e teoria degli insiemi - Conoscenze elementari dilinguaggi formali

Eventuali corsi propedeutici

- Logica e Matematica Discreta - Linguaggi Formali e

Compilatori

2. Obiettivi formativi:

Il corso mira ad offrire agli studenti gli strumenti formali per il ragionamento rigoroso sia in linguaggio naturale che in linguaggi formali. In particolare si vuole enfatizzare le potenzialita della logica come strumento per una descrizione rigorosa della specifica di un programma e per porre le basi della verifica automatica di proprietà di programmi.

3. Risultati dell'apprendimento attesi:

Lo studente ha acquisito i principi fondamentali della logica formale. Inoltre, ha compreso le idee alla base delle tecniche disponibili per il suo utilizzo per la specifica e la verifica dei programmi.

4. Modalità di verifica dell'apprendimento:

Esame orale

5. Modalità d'insegnamento:

Tradizionale

6. Attività di supporto:

Non previste

7. Programma:

Logica proposizionale:

: sistemi deduttivi, interpretazione e modelli, correttezza e completezza; risoluzione, BDD, SAT solver.

Logica al primo-ordine:

: sistemi deduttivi, interpretazione e modelli, correttezza e completezza; resolution and Prolog.

Cenni ad altre sistemi logici.

8. Testi consigliati e bibliografia:

Mathematical Logic for Computer Science (3rd Ed.) Ben-Ari, Mordechai. Springer-Verlag, 2012.

ISBN 978-1-4471-4129-7

https://link.springer.com/book/10.1007/978-1-4471-4129-7

Materiale a cura del docente verra distribuito nella pagina moodle del corso.

Insegnamento	MFN0578 - Matematica Discreta e logica

Insegnamento (inglese):	Discrete Mathematics and Logic
CFU:	12
Settore:	MAT/01 - LOGICA MATEMATICA MAT/02 - ALGEBRA
Periodo didattico:	1
Tipologia di Attività Formativa:	A - di base
Docenti:	Alessandro ANDRETTA (Titolare) Andrea MORI (Titolare) Luca MOTTO ROS (Titolare) Nadir MURRU (Titolare)

1. Prerequisiti e Propedeuticità:

Competenze attese in ingresso

Conoscenza delle basi della matematica della scuola

superiore: in particolare le operazioni aritmetiche di base, le proprietà delle potenze, le equazioni di primo e secondo grado. Conoscenza della terminologia di base relativa alle parti del discorso: nomi, verbi, proposizioni, aggettivi.

Eventuali corsi propedeutici

Nessuno

2. Obiettivi formativi:

L'insegnamento si propone di fornire allo studente una introduzione alla matematica discreta e alla logica matematica, con particolare riguardo per i loro aspetti piu` rilevanti per la formazione di base di un informatico, in particolare una adeguata familiarita` con le strutture algebriche, il calcolo combinatorio e le principali tecniche di dimostrazione.

3. Risultati dell'apprendimento attesi:

MATEMATICA DISCRETA. Al termine del corso lo studente dovrà da prova di aver compreso le basi fondamentali della teoria della teoria dei gruppi e degli anelli e dei loro morfismi, con la conoscenza di alcuni specifici esempi; in particolare ci si aspetta la conoscenza della struttura algebrica degli interi, con applicazione all'aritmetica modulare. Lo studente dovrà essere in grado di manipolare permutazioni, risolvere equazioni diofantee lineari in due variabili, congruenze lineari ed impostare e risolvere problemi di carattere enumerativo utilizzando in modo appropriato il linguaggio ed il formalismo della teoria degli insiemi, com particolare riguardo alle operazioni tra insiemi (unione, intersezione, prodotto), le relazioni e le funzioni.

LOGICA Al termine del corso lo studente dovra` dimostrare di sapere riconoscere le principali tecniche di dimostrazione (diretta, per assurdo, per contrapposizione, per casi) in semplici dimostrazioni di proposizioni relative a strutture algebriche e relazionali. Dovra` essere in grado di utilizzare le principali forme del principio di induzione (ordinaria, forte, principio del

minimo) in semplici dimostrazioni aritmetiche o relative alla sintassi formale dei linguaggi proposizionali e del prim'ordine. Dovra` dimostrare di essere in grado di formalizzare mediante formule logiche semplici asserzioni formulate in italiano. Dovrà essere in grado di valutare la validità o meno di semplici formule del primo ordine in strutture algebriche in accordo con le regole della semantica di Tarski e dovrà essere in grado di stabilire quando una formula non è conseguenza logica di altre formule provvedendo semplici strutture algebriche o relazionali che producono un controesempio. Dovra` inoltre dimostrare di conoscere la terminologia di base della teoria dei reticoli e delle algebre di Boole, con particolare attenzione ai concetti di atomo, insieme inferiore, distributività, assieme al loro utilizzo nella rappresentazione dei reticoli distributivi e delle algebre di Boole finite. Infine, lo studente dovra` dimostrare di riconoscere l'esistenza o meno di biiezioni tra insiemi

infiniti costruiti mediante le operazioni di unione disgiunta, prodotto cartesiano, insieme potenza, e insieme delle sequenze finite.

ENGLISH

DISCRETE MATHEMATICS At the end of the course the student will have to show an understanding of the basic structure of groups and rings and their morphisms, with knowledge of some specific examples; in particular the student is expected to know the algebraic structure of the integers, with applications to modular arithmetic. The student will have to be able to manipulate permutations, solve linear diophantine equations in two variables, linear congruences and set and solve enumerative problems using in a proper way the language and formalism of set theory, specifically using set operations (union, intersection, product), relations and functions.

LOGIC Once the course is completed the student must show a basic understanding of the main proof techniques employed in basic mathematical arguments (direct proof, proof by contraposition, proof by cases) at least for the case of basic arguments establishing elementary properties of algebraic and relational structures. The student must also be able to master the use of the induction principle in simple arguments regarding

arithmetic properties, or syntactic properties of propositional or first order languages. The student must also be able to formalize in first order logic simple assertions about arithmetic properties of natural or real numbers. The student must also be able to apply the rules of Tarski semantic in order to establish the truth or falsity of a given formula in a given structure, moreover he must also be able to establish when a given formula is not a logical consequence of other formulae providing simple relational or algebraic structures which give a counterexample. The student must also know the basic terminology for lattices and boolean algebras with a particular emphasis on the notions of distributivity, infimum of a subset of a lattice, and their use in the representation of finite lattices. Finally the student must have a basic familiarity with the set theoretic concepts of function,bijection, surjection, injection and must be able to recognize whether there exist or not a bijection between certain types of infinite sets.

4. Modalità di verifica dell'apprendimento:

ITALIANO L'esame consiste di due parti, una di Matematica Discreta e una di Logica, che possono essere affrontate anche in appelli differenti, tuttavia entro lo stesso anno accademico (dall'appello di Gennaio a quello di Settembre). Le prove di esame sono scritte, a meno di motivata richiesta di esame orale da parte dei docenti. Le domande possono riguardare sia la teoria che lo svolgimento di esercizi. L'esito dell'esame e` espresso in trentesimi come media dei punteggi riportati nelle parti di Matematica Discreta e di Logica.

ENGLISH. The exam consists of two independent parts, Discrete Mathematics and Logic, that can be taken in different sessions but always within the same Academic Year (from January to September). The two parts of the exams are written, unless the teachers require an oral exam. The questions in the exams can either be theoretical or problem-solving in nature. The final grade of the exam is obtained as the average of the two parts, expressed as a fraction of 30.

5. Modalità d'insegnamento:

ITALIANO Lezioni frontali (52 ore Matematica Discreta e 52 ore Logica) comprese le esercitazioni in aula.

ENGLISH Lectures (52 hrs Discrete Mathematics + 52 hrs Logic) including exercises in classroom.

6. Attività di supporto:

ITALIANO Disponibilita` di un tutor.

ENGLISH A tutor will be available.

7. Programma:

Linguaggio degli insiemi (14 ore circa) • Insiemi: insieme vuoto; sottoinsiemi; unione; intersezione; complementare; insieme delle parti (con particolare attenzione al caso finito). • Corrispondenze, relazioni e funzioni: relazioni d'ordine. • Relazioni di equivalenza e partizioni. • Composizione e inversione di corrispondenze. • Iniettivita`, suriettivita`, composizione e invertibilita` di funzioni.

Calcolo combinatorio (12 ore circa) • Cardinalita` di insiemi finiti • Principi della somma e del prodotto • Disposizioni semplici e con ripetizioni • Combinazioni semplici e con ripetizioni. • Il Teorema del binomio e il triangolo di Pascal-Tartaglia • Il principio di inclusione-esclusione

Strutture algebriche (10 ore circa) • Semigruppi e loro morfismi. • Monoidi e loro morfismi: monoide delle parole • Gruppi e loro morfismi. • Alcuni esempi di strutture algebriche: numeri naturali e interi, gruppo delle biiezioni di un insieme. • Gruppi e sottogruppi ciclici. • Sottogruppi e Teorema di Lagrange. • Anelli: anello delle Matrici. • Corpi e campi: campo dei numeri razionali.

Aritmetica modulare (8 ore circa) • Anelli degli interi e delle classi di resto. • Teorema della divisione • L'algoritmo di Euclide • Identità di Bezout • Equazioni diofantee • Il teorema di EuleroFermat

Gruppo delle permutazioni (8 ore circa) • Composizione, potenze e inverse di permutazioni. • Decomposizione in cicli

disgiunti e decomposizione in trasposizioni. • Parita` di una permutazione • Sottogruppi del gruppo delle permutazioni.

Logica

Tecniche di dimostrazione (8 ore circa) • Dimostrazione diretta, per assurdo, per contrapposizione;

• Connettivi logici e loro significato in termini di condizioni di verita`; • Tavole di verita` e conseguenza logica tra proposizioni.

Il principio di induzione (6 ore circa) • Forma ordinaria e forte del principio di induzione; • Principio del minimo; • Equivalenza tra forme del principio di induzione; • Induzione strutturale; • Ricorsione.

Formalizzazione (8 ore circa) • Linguaggi proposizionali e del prim'ordine: termini, quantificatori, alfabeto non logico, formule; • Schemi di traduzione da linguaggio naturale in linguaggi del prim'ordine (condizione sufficiente, necessaria, per tutti gli n abbastanza grandi, ci sono n arbitrariamente grandi,...)

Semantica della Logica del primo ordine (14 ore circa): Regole della semantica di Tarski per

formule del primo ordine: esempi, e definizioni — Conseguenza logica tra formule del primo ordine: definizioni e costruzione di contro-esempi

Reticoli (6 ore circa) • Ordinamenti parziali, con esempi; • Minimo confine superiore, massimo confine inferiore: reticoli; • Distributivita`;

Algebre di Boole (6 ore circa) • Algebra degli insiemi; • Algebre di Boole: definizioni; • Algebre di Boole come reticoli; • Cenni alla rappresentazione delle algebre di Boole finite: atomi; • Calcolo delle funzioni booleane e algebra della commutazione; • Algebra della logica: algebre di Boole e calcolo proposizionale.

Insiemi infiniti (4 ore circa) • Insiemi numerabili e piu` che numerabili: esempi; • Operazioni infinitarie: unioni e intersezioni, prodotti e somme, con le principali proprieta`.

8. Testi consigliati e bibliografia:

A. Mori, Lezioni di Matematica Discreta.

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

C. Bertone, M. Roggero, *Appunti ed Esercizi di Matematica Discreta*, 2017/18.

F. Preparata, R.T. Yeh, *Introduzione alle strutture discrete*, Boringhieri, 1976.

Andretta, Cardone, Motto Ros, Viale: *Dispense di logica matematica*, 2017.

Insegnamento	**MFN0633 - Metodi Formali dell'Informatica**

Insegnamento (inglese):	**Formal Methods in Computer Science**
CFU:	**9**
Settore:	**INF/01 - INFORMATICA**
Periodo didattico:	**1** **2**

Tipologia di Attività Formativa: **B - caratterizzante**

Docenti: **Ugo DE' LIGUORO (Titolare)**

---

1. Prerequisiti e Propedeuticità:

Competenze attese in ingresso

Sono richieste conoscenze elementari di logica e buone conoscenze di programmazione e algoritmi; inoltre si assume che lo studente possegga nozioni di linguaggi formali.

Eventuali corsi propedeutici

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

2. Obiettivi formativi:

Il corso si propone di offrire agli studenti le conoscenze di base relative alla verifica formale di proprietà dei programmi. A questo scopo vengono introdotte le basi

teoriche della semantica dei programmi e della logica di Floyd-Hoare. Inoltre saranno introdotte tecniche di analisi statica di programmi con puntatori basate sulla separation logic.

Il corso introduce all'uso di proof-assistant per l'analisi di programmi assistita da calcolatore, quali HOL/Isabelle e VeriFast.

3. Risultati dell'apprendimento attesi:

Conoscenza di alcune tecniche per la verifica formale di programmi basate sulla logica e sulla specifica mediante "contratti". Abilità nel costruire dimostrazioni formali di correttezza dei programmi mediante l'uso di un tool. Consapevolezza della necessità di sviluppare il codice in modo ben strutturato e di documentarlo in modo formalmente preciso per facilitarne la verifica ed assicurarne l'affidabilità.

4. Modalità di verifica dell'apprendimento:

L'esame consiste in un esame orale, in cui saranno verificate non solo le conoscenze teoriche, ma anche le

abilità pratiche nell'uso dei tool di verifica interattiva utilizzati durante il corso.

5. Modalità d'insegnamento:

Le lezioni si svolgono in aula sia in modo tradizionale che interattivo mediante esercitazioni. Si farà ampio uso della piattaforma Moodle per la proposta di esercizi e per la discussione del loro svolgimento.

6. Attività di supporto:

Il materiale didattico di supporto (a cura del docente) si può reperire nella pagina Moodle del corso stesso.

7. Programma:

- Formalizzazione e proof assistants
    - Introduzione ad HOL/Isabel: programmazione e dimostrazioni
        - Logica equazionale e predicativa in Isabel
        - Induzione matematica e strutturale Uso di Isar
    - IMP: semantica di un semplice linguaggio imperativo
        - Semantica operazionale big-step
        - Semantica operazionale one-step
    - Equivalenza delle semantiche big-step e one-step
    - Analisi statica
        - Correttezza di un compilatore per IMP
        - Tipi per IMP ed information-flow analysis
        - Data-flow ed ottimizzazione
    - Logica dei programmi
        - Logica del primo ordine
        - Asserzioni e logica di Floyd-Hoare
        - Automazione della verifica dei programmi, verification conditions
        - Calcolo degli updates e verifica con KeY-Hoare

- Analisi dei puntatori e separation logic
  - Separation logic
  - Definizione formale di VeriFast
  - Verifica di programmi C con VeriFast

8. Testi consigliati e bibliografia:

- G. Winskel: "La semantica formale dei linguaggi di programmazione", The MIT Press, edizioneitaliana UTET, 1999.
- T. Nipkow, G. Klein: "Concrete Semantics", Springer 2014.

Ulteriori riferimenti ad articoli e dispense del docente saranno forniti attraverso la piattaforma Moodle.

Insegnamento	**INF0190 - Metodi Formali dell'Informatica Parte A**
Insegnamento	**Formal Methods in Computer Science - A**

(inglese):	
CFU:	**6**
Settore:	**INF/01 - INFORMATICA**
Periodo didattico:	**1**
	2
Tipologia di Attività Formativa:	**D - libera**
Docenti:	**Ugo DE' LIGUORO (Titolare)**

1. Prerequisiti e Propedeuticità:

Competenze attese in ingresso

Sono richieste conoscenze elementari di logica e buone conoscenze di programmazione e algoritmi; inoltre si assume che lo studente possegga nozioni di linguaggi formali.

Eventuali corsi propedeutici

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

2. Obiettivi formativi:

Il corso si propone di offrire agli studenti le conoscenze di base relative alla verifica formale di proprietà dei programmi. A questo scopo vengono introdotte le basi teoriche della semantica dei programmi e della logica di Floyd-Hoare. Inoltre saranno introdotte tecniche di analisi statica di programmi con puntatori basate sulla separation logic.

Il corso introduce all'uso di proof-assistant per l'analisi di programmi assistita da calcolatore, quali HOL/Isabelle e VeriFast.

3. Risultati dell'apprendimento attesi:

Conoscenza di alcune tecniche per la verifica formale di programmi basate sulla logica e sulla specifica mediante "contratti". Abilità nel costruire dimostrazioni formali di correttezza dei programmi mediante l'uso di un tool. Consapevolezza della necessità di sviluppare il codice in modo ben strutturato e di documentarlo in modo formalmente preciso per facilitarne la verifica ed assicurarne l'affidabilità.

4. Modalità di verifica dell'apprendimento:

L'esame consiste in un esame orale, in cui saranno verificate non solo le conoscenze teoriche, ma anche le abilità pratiche nell'uso dei tool di verifica interattiva utilizzati durante il corso.

5. Modalità d'insegnamento:

Le lezioni si svolgono in aula sia in modo tradizionale che interattivo mediante esercitazioni. Si farà ampio uso della piattaforma Moodle per la proposta di esercizi e per la discussione del loro svolgimento.

6. Attività di supporto:

Il materiale didattico di supporto (a cura del docente) si può reperire nella pagina Moodle del corso stesso.

7. Programma:

- Formalizzazione e proof assistants
  - Introduzione ad HOL/Isabel: programmazione e dimostrazioni
    - Logica equazionale e predicativa in Isabel
    - Induzione matematica e strutturale Uso di Isar
  - IMP: semantica di un semplice linguaggio imperativo
    - Semantica operazionale big-step
    - Semantica operazionale one-step
  - Equivalenza delle semantiche big-step e one-step
  - Analisi statica
    - Correttezza di un compilatore per IMP
    - Tipi per IMP ed information-flow analysis
    - Data-flow ed ottimizzazione
  - Logica dei programmi
    - Logica del primo ordine
    - Asserzioni e logica di Floyd-Hoare
    - Automazione della verifica dei programmi, verification conditions

8. Testi consigliati e bibliografia:

- G. Winskel: "La semantica formale dei linguaggi di programmazione", The MIT Press, edizioneitaliana UTET, 1999.
- T. Nipkow, G. Klein: "Concrete Semantics", Springer 2014.

Ulteriori riferimenti ad articoli e dispense del docente saranno forniti attraverso la piattaforma Moodle.

Insegnamento	**INF0193 - Metodologie e Tecnologie Didattiche per l'Informatica (PREFIT)**

Insegnamento (inglese):	**Methodologies and technologies for teaching informatics (PREFIT)**
CFU:	**6**
Settore:	**INF/01 - INFORMATICA**
Periodo didattico:	**2**
Tipologia di Attività Formativa:	**D - libera**
Docenti:	**Giuseppina Barbara DEMO (Titolare)**

1. Prerequisiti e Propedeuticità:

Competenze attese in ingresso

Buona capacità di programmazione, conoscenza di algoritmi fondamentali e dimestichezza coi sistemi di gestione di basi di dati e con la modellizzazione della realtà nel modello Entità Relazioni.

2. Obiettivi formativi:

- Sviluppare capacità riflessive sulla disciplina, facendo emergere per confrontare le idee pregresse degli studenti sugli argomenti trattati. - Conoscenza delle principali metodologie per la costruzione di un curriculum di Informatica coerente con gli obiettivi fissati dalle indicazioni nazionali e dalle linee guida. - Sperimentazione di attività laboratoriali per lo sviluppo e la valutazione di competenze e riconoscimento della loro importanza - Acquisizione di pratiche didattiche e di processi di insegnamento e apprendimento dell'Informatica sia con l'uso delle tecnologie digitali sia con tecniche di tipo csunplugged ovvero senza calcolatore con attività per i diversi livelli di scuola - Riconoscimento della stretta relazione esistente tra l'Informatica e temi rilevanti per la società.

3. Risultati dell'apprendimento attesi:

Padronanza delle metodologie e delle tecniche necessarie per l'insegnamento del pensiero computazionale e dell'organizzazione dell'informazione.

Conoscenza approfondita delle pratiche didattiche di insegnamento dell'Informatica intesa come scienza dell'informazione. Capacità di descrivere e declinare le relazioni esistenti tra Informatica e temi rilevanti per la società.

4. Modalità di verifica dell'apprendimento:

La verifica finale si svolgerà, di norma, come segue: - Discussione di una attività, da sviluppare in aula in Python o in altro linguaggio, ideata e realizzata dallo studente, o di un progetto in
Greenfoot, da concordare. - Progetto di unità didattica da svolgere presso una scuola secondaria o in un corso di aggiornamento per docenti di qualunque livello e tipo di scuola, da concordare Discussione sui temi affrontati nel corso.

5. Modalità d'insegnamento:

Il corso propone esempi di attività di programmazione nonchè di organizzazione e ricerca di informazioni riproponibili nel contesto della scuola secondaria di

secondo grado. Le esperienze inducono a riconoscere nell'ambiente quotidiano, nelle discipline tradizionali, negli strumenti di comunicazione elementi fondamentali dell'Informatica allo scopo di imparare ad utilizzarli nell'insegnamento e a coglierne le potenzialità didattiche.

6. Attività di supporto:

7. Programma:

Analisi delle principali metodologie per l'insegnamento dell'informatica presenti in letteratura ponendo in particolare l'accento sulla distinzione fra l'informatica come disciplina scientifica da una parte e le applicazioni dell'informatica dall'altra. Quindi richiami terminologici a cominciare dalla distinzione fra informatica e alfabetizzazione digitale generale e specialistica.

Processi di insegnamento e apprendimento dell'Informatica con e senza l'uso delle tecnologie. In particolare analisi di metodologie e tecniche di dattiche

quali: -csunplugged ovvero didattica dell'informatica senza calcolatore -costruzionismo di Papert e robotica educativa.

Presentazione di ambienti di supporto alla didattica della programmazione. In particolare Introduzione agli ambienti Scratch e Greenfoot e loro analisi secondo le caratteristiche proposte nella letteratura, in particolare da Romeike e Schneiderman --Linguaggi di programmazione testuali: Python. Analisi dell'efficacia di strumenti didattici multimediali per lo studio dell'Informatica.

Parallelismo fra metodologie informatiche e metodologie didattiche: pensiero computazionale nell'affrontare i problemi visto in diversi settori dell'informatica (nella programmazione, nell'interrogare una base di dati e nel csunplugged); ambienti di programmazione e pedagogia dell'imparare ad imparare e imparare collaborando (cooperative learning) nello sviluppo e realizzazione di algoritmi in vari ambienti digitali e non.

Modellizzazione della realtà e rappresentazione della conoscenza, interrogazione di una base di dati.

Ruolo delle proposte Bebras e Olimpiadi del Problem

solving nella didattica dell'informatica.

Strumenti di elearning e sistemi social per gli insegnanti

Educazione all'uso consapevole dei social media.

8. Testi consigliati e bibliografia:

Il materiale didattico sarà disponibile nel sito internet http://informatica.ilearn.unito.it/course/index.php?categ testi base consigliati per il corso sono dispense e appunti f docenti. Materiali indicati per gli approfondimenti ed integ sono: Académie des Sciences, L'enseignement de l'informat France - Il est urgent de ne plus attendre, maggio 2013, http://www.academie-sciences.fr/activite/rapport/rads_0 Bell,T.,Witten, I.H.andFellows, M., CS Unplugged, 1998, http://csunplugged.org - Miles Berry, "Computing in the N

Curriculum. Guide for primary teachers", Computing at scho
http://www.computingatschool.org.uk/data/uploads/CASPr
uting.pdf - Gander. D. et al.,
Informatics Education. Europe cannot afford to miss the bo
Report, Kogan Page, London,
2013, http://europe.acm.org/iereport/index.html - Guzdia
Ericson, B. Introduction to
Computing and Programming in Python, A Multimedia Appro
Pearson, 2010 - Indicazioni Nazionali per il curricolo della
dell'Infanzia e del primo ciclo d'istruzione, novembre 2012
http://hubmiur.pubblica.istruzione.it/web/istruzione/prot7
M. Lodi, Imparare il pensiero computazionale - imparare a
programmare, Tesi magistrale, Università di Bologna, 2013.
dell'Informatica per la scuola (secondaria), Gruppo di lavo
maggio 2013,
//www.grin-
informatica.it/opencms/opencms/grin/infoescuola/riformas
Resnick M., Maloney
J., Monroy-Hernandez A., Rusk N., Eastmond E., Resnick M
K., Millner A., Rosenbaum E.,
Silver J., Silverman B., & Kafai Y., Scratch: Programming
Communications of the ACM,
Vol.52 n.11, novembre 2009 - R. Romeike, Three Drivers f
Creativity in Computer Science

Education. In: Benzie, D.; Iding, M. (eds.): Proceedings o
Conference on "Informatics,
Mathematics and ICT: A golden triangle", June 27-29, 200
USA. - B. Schneiderman, 2002, B. Schneiderman, Creativit
tools. Communications of the ACM, Vol. 45, Issue 10 (Oct.
C. Schulte, Reflections on the role of programming in prima
secondary computing education, Proceeding WiPSE '13 Proc
the 8th Workshop in Primary and Secondary Computing Educ
2013 - B. Vertecchi , Contrordine in classe "Attenti al tabl
nuovi analfabeti" , La Repubblica 7/1/2016,
http://ricerca.repubblica.it/repubblica/archivio/repubblica
16/01/07/contrordine-in-classe-attenti-al-tablet-crea-nuovi-
analfabeti21.html - Jeannette Wing, Computational thinking,
Communications of the ACM, Vol. 49, Issue 3, (marzo 2006).

Insegnamento	MFN0582 - Programmazione I

Insegnamento (inglese):	Programming I
CFU:	9
Settore:	INF/01 - INFORMATICA
Periodo didattico:	1
Tipologia di Attività Formativa:	A - di base
Docenti:	Elvio Gilberto AMPARORE (Titolare) Felice CARDONE (Titolare) Valentina GLIOZZI (Titolare) Alessandro MAZZEI (Titolare) Amon RAPP (Titolare) Luca ROVERSI (Titolare) Giancarlo Francesco RUFFO (Titolare)

1. Prerequisiti e Propedeuticità:

Competenze attese in ingresso

Non è richiesto alcun prerequisito specifico alla programmazione. È consigliabile avere capacita' di uso del calcolatore con sistema a finestre. È invece opportuno possedere conoscenze di base, quali i concetti di numero (naturale, intero, razionale, reale), di funzione, le quattro operazioni, elevamento a potenza, radice, esponenziale, logaritmo, piano cartesiano, calcolo algebrico elementare. Infine, è indispensabile una buona padronanza della lingua madre, possibilmente accompagnata da una propensione al ragionamento strutturato.

Eventuali corsi propedeutici

Nessuno.

2. Obiettivi formativi:

Fornire i concetti di base della programmazione strutturata ed imperativa di alto livello, appoggiandosi ad un linguaggio di programmazione di riferimento.

3. Risultati dell'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 al punto (i) in un linguaggio di programmazione di riferimento; (iii) valutare correttezza parziale e terminazione di procedure per mezzo di semplici dimostrazioni formali; (iv) valutare l'efficienza di una procedura; (v) rappresentare istantanee dello stato in cui si trovi la memoria di un calcolatore che interpreti programmi ragionevolmente complessi nel linguaggio di programmazione di riferimento.

4. Modalità di verifica dell'apprendimento:

Sono impostate per verificare conoscenza e comprensione degli argomenti sviluppati nel programma didattico, in accordo con i "Descrittori di Dublino", relativi ai learning outcomes. Per mezzo di un numero congruo di esercizi saranno verificate conoscenze e capacità di:

- progettare algoritmi iterativi e ricorsivi;

codificare programmi che implementano algoritmi iterativi e ricorsivi;

- risolvere esercizi sia su aspetti fondamentali della programmazione di alto livello (correttezza, terminazione e costo degli algoritmi) sia sulla simulazione della gestione della memoria.

Gli esercizi saranno scritti o al calcolatore. La somma dei punteggi degli esercizi proposti permetterà di ottenere voti sino al "30 e Lode".

5. Modalità d'insegnamento:

Le lezioni in aula sono svolte alla lavagna (virtuale o meno) e, se opportuno, con l'ausilio di calcolatore.

Il laboratorio e' organizzato a turni. Il numero di turni puo' variare di anno in anno a seconda del numero di iscritti. E' usuale avere due turni per il corso A e due per il corso B.

Istruzioni su come accedere al laboratorio, saranno comunicate in aula, durante le prime lezioni, e rese disponibili sul sito di supporto alla didattica.

6. Attività di supporto:

Sono previste 12 ore facoltative di esercitazione in aula in cui saranno proposti esercizi e relative soluzioni sugli argomenti principali del corso. On-line saranno disponibili:

- dispense su argomenti fondamentali del corso,
- non esaurientemente coperti dal testo di riferimento; documentazione disponibile su Internet; software.

7. Programma:

Struttura

- di base di un calcolatore;
- Informazioni essenziali su linguaggi di programmazione, differenze tra 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, modello di gestione della memoria;

- Correttezza parziale, terminazione e nozioni di costo degli

algoritmi.

8. Testi consigliati e bibliografia:

Walter

- Savitch "Programmazione di base e avanzata con Java", Pearson --- ISBN 9788865181904.
- Dispense e lucidi ufficiali, distribuiti dai docenti.

Insegnamento	**MFN0585 - Programmazione II**
Insegnamento (inglese):	**Programming II**
CFU:	**9**
Settore:	**INF/01 - INFORMATICA**
Periodo didattico:	**2**
Tipologia di Attività Formativa:	**A - di base**
Docenti:	**Elvio Gilberto AMPARORE (Titolare)** **Stefano BERARDI (Titolare)** **Viviana BONO (Titolare)** **Ferruccio DAMIANI (Titolare)**

Diego MAGRO (Titolare)
Rossano SCHIFANELLA (Titolare)
Gianluca TORTA (Titolare)

1. Prerequisiti e Propedeuticità:

Competenze attese in ingresso

Agli studenti è richiesta la conoscenza delle nozioni fondamentali di programmazione imperativa nel linguaggio Java (comandi di assegnamento, condizionali e iterativi, tipi di dato semplici e array, nozioni di astrazione procedurale e ricorsione).

Eventuali corsi propedeutici

Programmazione I e Laboratorio

2. Obiettivi formativi:

L'insegnamento ha l'obiettivo di approfondire concetti di informatica di base e in particolare di fornire una introduzione al paradigma di programmazione a oggetti. Si propone quindi di raffinare le capacità di

programmare nel linguaggio Java apprese nel corso di Programmazione I e di introdurre le nozioni fondamentali della programmazione orientata agli oggetti. In particolare, il corso illustrerà le *astrazioni fondamentali* per la progettazione del software (classi e oggetti), la definizione di *semplici strutture dati* (liste, alberi, pile, code) e operazioni corrispondenti, i meccanismi di base per favorire *riuso e modularità* del software (ereditarietà, polimorfismo, tipi generici), la specifica degli *invarianti di classe* e gestione delle loro violazioni (asserzioni ed eccezioni), così come alcune *classi fondamentali* della libreria Java. Si darà particolare enfasi agli aspetti di buona progettazione del software, utilizzando concetti presi a prestito dall'ingegneria del software e formalismi grafici quali UML.

3. Risultati dell'apprendimento attesi:

Lo studente acquisirà la capacità di progettare e realizzare piccole/medie applicazioni in Java sfruttando gli strumenti di base della programmazione a oggetti, imparerà a utilizzare ed a realizzare alcune strutture dati fondamentali e approfondirà la conoscenza della

libreria standard di Java.

4. Modalità di verifica dell'apprendimento:

L'esame si compone di una verifica di laboratorio e di una prova scritta.

La *verifica di laboratorio* consiste in una prova pratica di programmazione al calcolatore della durata di 45 minuti, viene valutata con un punteggio da 0 a 2 e considerata sufficiente se tale punteggio è 1 o 2. Una volta superata la verifica di laboratorio, il punteggio ottenuto rimarrà valido fino all'appello precedente l'avvio del corso dell'anno successivo. Uno studente può consegnare al massimo 3 elaborati di laboratorio durante l'anno accademico. Può però presentarsi ad ogni appello (previa registrazione) e decidere se consegnare o meno il proprio elaborato.

La *prova scritta*, che può essere sostenuta solo previo superamento della verifica di laboratorio, consiste in esercizi di programmazione e sulle nozioni fondamentali del corso e viene valutata con un punteggio da 0 a 30. È considerata sufficiente se tale punteggio è 18 o superiore.

Il voto finale dell'esame di Programmazione II e Laboratorio è pari al punteggio della prova scritta.
La lode viene assegnata se tale punteggio è 30 e il punteggio della prova di laboratorio è 2.

5. Modalità d'insegnamento:

L'insegnamento consta di una parte teorica (60 ore) e una pratica (30 ore).

La parte teorica si svolge in aula con l'ausilio di lavagna e gesso ed eventualmente di un calcolatore usato dal docente per la proiezione di lucidi e/o lo sviluppo di semplici programmi.

La parte pratica si svolge nei laboratori didattici in cui il docente illustra esercizi di programmazione che gli studenti risolvono al calcolatore. Il docente (ed eventualmente studenti tutor) supportano gli studenti nella risoluzione degli esercizi assegnati.

6. Attività di supporto:

Il materiale usato durante le lezioni in aula e in laboratorio, i testi degli esercizi e delle prove d'esame

sono documentati e/o resi disponibili sulla pagina moodle del corso http://informatica.ilearn.unito.it/.

7. Programma:

Il programma del corso è basato sul testo di riferimento, eventualmente integrato da dispense e/o altro materiale fornito dal docente e reso disponibile sulla pagina moodle del corso. Segue un sommario degli argomenti trattati e dei capitoli relativi sul testo di riferimento. Tale sommario non riflette necessariamente l'ordine di esposizione degli argomenti.

- ripasso dei concetti di base della programmazione imperativa e di Java (Cap. 1-7) incapsulamento, definizione di una classe, creazione e uso di oggetti (Cap. 8 e 9) specializzazione di una classe, ereditarietà (Cap. 10)
relazioni tra classi, polimorfismo, classi parzialmente definite, interfacce (Cap. 11) generalizzazione di una classe, classi parametriche, tipi generici (Cap. 12) invarianti di classe, asserzioni, eccezioni e loro gestione (Cap. 13)

progettazione e implementazione di strutture dati (liste, alberi, pile, code) (Cap. 15) collezioni e iteratori (Cap. 16) cenni su progettazione a oggetti e UML

8. Testi consigliati e bibliografia:

Libro di testo:

- W. Savitch: Programmazione di base e avanzata con Java, Seconda edizione, Pearson, 2018 (anche la prima edizione va bene).

Altri testi di riferimento:

- M. Naftalin, P. Wadler: Java Generics and Collections, O'Reilly, 2006.

G. Pighizzini, M. Ferrari: Dai fondamenti agli oggetti, Terza edizione, Pearson Addison Wesley, 2008.

Insegnamento	**MFN0605 - Programmazione III**
Insegnamento	**Programming III**

(inglese):	
CFU:	**6**
Settore:	**INF/01 - INFORMATICA**
Periodo didattico:	**1**
Tipologia di Attività Formativa:	**B - caratterizzante**
Docenti:	**Liliana ARDISSONO (Titolare)**

1. Prerequisiti e Propedeuticità:

Competenze attese in ingresso

Si richiede una buona conoscenza dell'analisi e della progettazione object-oriented (fornita dagli insegnamenti Programmazione II e Algoritmi e strutture dati) e dei meccanismi di base e delle problematiche della programmazione concorrente (fornita dall'insegnamento Sistemi Operativi). Gli studenti e le studentesse devono inoltre avere la capacità di scrivere, compilare e verificare la correttezza di programmi in Java.

Eventuali corsi propedeutici

Programmazione II, Algoritmi e strutture dati, Sistemi Operativi.

2. Obiettivi formativi:

Questo insegnamento si colloca nell'area di approfondimento del percorso formativo del Corso di Laurea in Informatica e ha come obiettivi formativi specifici l'insegnamento dei concetti fondamentali e delle metodologie della programmazione distribuita e concorrente.

Lo sviluppo di software efficiente e scalabile presuppone la capacità di programmare applicazioni distribuite e concorrenti. In particolare, la programmazione distribuita in ambiente object oriented arricchisce le nozioni di base di programmazione concorrente sfruttando il paradigma ad oggetti per una più chiara scomposizione delle attività da eseguire in parallelo e loro attribuzione alle entità software di competenza, che possono essere modellate come oggetti distribuiti che

offrono i relativi servizi.

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

In questo insegnamento, le capacita' di applicare le conoscenze acquisite vengono sviluppate nell'ambito di una parte di laboratorio che permette di sperimentare le nozioni apprese durante le ore di teoria in aula, organizzando il lavoro in piccoli gruppi (max 3 persone). I temi introdotti durante il laboratorio corredano e integrano le conoscenze derivanti dalla

parte teorica
(knowledge and understanding) e permettono agli studenti e alle studentesse di familiarizzare con le metodologie e tecnologie introdotte, e di analizzare soluzioni alternative (applying knowledge and understanding). Inoltre nell'ambito delle attivita' di laboratorio è previsto lo sviluppo di un'applicazione distribuita realistica, con interfaccia grafica, che va oltre la complessita' del tipico esercizio svolto durante le ore di lezione. Lo svolgimento del progetto di laboratorio permette di allenare l'autonomia di giuzio nella scelta delle implentazioni da realizzare, analizzando in forma critica le informazioni acquisite nella parte teorica dell'insegnamento. La preparazione del progetto di laboratorio e' anche volta a stimolare la capacità comunicativa degli studenti e delle studentesse che sono invitati/e a illustrare verbalmente le soluzioni adottate (communication skills).

3. Risultati dell'apprendimento attesi:

Gli studenti e le studentesse acquisiranno la conoscenza dei concetti che stanno alla base della programmazione

di interfacce utente grafiche e di applicazioni distribuite e concorrenti in linguaggi ad alto livello. Essi/esse dovranno essere in grado di sviluppare applicazioni distribuite in ambiente Java, utilizzando i Socket Java per la realizzazione di applicazioni distribuite, la programmazione a thread per la gestione del parallelismo e SWING e JavaFX per la realizzazione delle interfacce utente grafiche.

4. Modalità di verifica dell'apprendimento:

L'esame è composto da un TEST SCRITTO e da una VERIFICA DI LABORATORIO. Il primo e' volto a verificare che gli studenti e le studentesse abbiano acquisito la conoscenza fornita dall'insegnamento in termini di sapere. Il secondo e' volto a verificare le competenze acquisite (saper fare), nonche' le abilita' comunicative e l'autonomia di giudizio. Le due prove possono essere sostenute in qualsiasi ordine.

- TEST SCRITTO: Prova scritta che include esercizi e domande teoriche sul programma dell'insegnamento. Viene valutata da un minimo di 0 ad un massimo di 30 e si considerano sufficienti i voti >=18. Come

da regolamento di Ateneo, ogni studente/studentessa può sostenere un numero massimo di tre prove scritte durante l'Anno Accademico (cioè, consegnare il proprio elaborato tre volte). Il voto ottenuto durante un test scritto decade se si partecipa ad un altro test scritto e si consegna l'elaborato.

. VERIFICA DI LABORATORIO: prevede la discussione del progetto di laboratorio svolto durante il corso. La discussione deve essere effettuata preferibilmente in unica soluzione, con tutti i membri del gruppo di laboratorio presenti, e si può tenere sia durante gli appelli appositamente dedicati che su appuntamento (previa email al/alla docente). Il voto di laboratorio è un numero compreso tra 0 e 30, si considerano sufficienti i voti >=18. . CALCOLO DEL VOTO FINALE DI ESAME: sia X il voto del test scritto; sia Y il voto di laboratorio. Il voto Fin finale dell'esame si ottiene come segue: Fin = (X+Y)/2

Note: i voti acquisiti durante la prova di laboratorio, o durante il test scritto, rimangono validi fino al termine della terza sessione d'esame (quella che precede l'inizio del nuovo insegnamento). Quando si superano entrambe

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

5. Modalità d'insegnamento:

L'insegnamento è diviso in una parte di teoria e una di laboratorio. Per la parte di teoria sono previste lezioni frontali che seguono il programma riportato più avanti, integrate da casi di studio e da esercitazioni volte ad illustrare l'applicazione pratica dei concetti appena studiati. La parte di laboratorio ha per oggetto l'insegnamento delle tecnologie per lo sviluppo di applicazioni
distribuite e concorrenti in Java (Java Socket, Thread), con interfacce grafiche a finestre (SWING e JavaFX/FXML). Le lezioni si svolgono in maniera

interattiva e sono corredate da vari esercizi miranti a fornire esempi pratici.

Le sperimentazioni che vengono effettuate durante le ore di laboratorio, strutturate come sequenze di esercizi specifici, sono fondamentali per aiutare gli studenti e le studentesse a comprendere e assimilare i contenuti teorici spiegati a lezione in quanto permettono di mettere in pratica i concetti e le metodologie illustrate su esempi concreti. Inoltre lo sviluppo del progetto di laboratorio, organizzato in gruppi di max 3 studenti/studentesse, permette di consolidare le conoscenze teoriche in un caso realistico di media complessità e di esercitarsi nel lavoro collaborativo.

Si consiglia caldamente la frequenza costante alle lezioni di teoria e di laboratorio. E' inoltre fondamentale iscriversi al corso online su I-Learn, all'interno del quale viene messo a disposizione materiale didattico di supporto.

6. Attività di supporto:

Il materiale didattico di supporto (lucidi delle lezioni, link a documentazione, esempi di testo di

esami ed altro) è disponibile presso il supporto on-line ai corsi: qui

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

7. Programma:

Programmazione ad eventi in Java: programmare interfacce grafiche.

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

Programmazione Multithread:

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

Programmazione in rete in Java:

- L'architettura client-server.

- Uso di socket.
- Polimorfismo e trasferimento di oggetti mediante Java.
- Il modello di esecuzione distribuita di oggetti.

8. Testi consigliati e bibliografia:

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

Altri testi (approfondimenti):

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

Insegnamento	**MFN0635 - Reti di Elaboratori**
Insegnamento (inglese):	**Computer Networks**

CFU:	**12**
Settore:	**INF/01 - INFORMATICA**
Periodo didattico:	**1**
	2
Tipologia di Attività Formativa:	**B - caratterizzante**
Docenti:	**Michele GARETTO (Titolare)** **Matteo SERENO (Titolare)**

---

1. Prerequisiti e Propedeuticità:

Competenze attese in ingresso

- Competenza nella programmazione sequenziale C
- e/o Java (per recepire gli elementi di programmazione concorrente e distribuita presentati nel corso);

Competenza nelle architetture degli elaboratori e dei sistemi operativi (per apprezzare gli aspetti sistemistici dei sistemi distribuiti oggetto di studio).

Eventuali corsi propedeutici

Programmazione I e II, Architetture, Sistemi Operativi.

2. Obiettivi formativi:

Il corso studia gli elementi fondamentali delle tecnologie di trasmissione del livello data link, dei protocolli di accesso a mezzi condivisi e dei protocolli di trasmissione wireless, la suite di protocolli TCP/IP, e i principi che guidano la strutturazione e la progettazione di applicazioni distribuite.

3. Risultati dell'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;
- Principi che guidano la strutturazione e la progettazione di applicazioni distribuite.

4. Modalità di verifica dell'apprendimento:

La verifica si articolerà in due prove scritte che

potranno richiedere complessivamente tra le tre e le quattro ore. La verifica sarà orientata ad accertare la capacità maturata dallo studente nella comprensione dei fenomeni e dei protocolli di livello data link, di accesso a mezzo condiviso, di trasmissione wireless, 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.

5. Modalità d'insegnamento:

Le lezioni e le esercitazioni sono svolte con l'ausilio di slide che verranno messe a disposizione sul sito di e-learning del corso di laurea.

6. Attività di supporto:

Il materiale didattico di supporto è disponibile sul siti e-learning del Corso di Laurea.

7. Programma:

- Introduzione ad Internet ed alle reti di calcolatori
- Il livello di collegamento e le reti locali:
- 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

8. Testi consigliati e bibliografia:

Libri di testo:

- Reti di Calcolatori e Internet - Un approccio top-down, J. Kurose, K. Ross, Pearson Education, Sesta Edizione.

Introduzione alla programmazione client-server, D. Maggiorini, Pearson, Addison Wesley

Libri consigliati:

- Internetworking con TCP/IP. Vol. 1: Principi,

protocolli e architetture, D. Comer, Pearson Prentice Hall

Insegnamento	**MFN1362 - Reti I**

Insegnamento (inglese):	**Computer Networks I**
CFU:	**6**
Settore:	**INF/01 - INFORMATICA**
Periodo didattico:	**1**
Tipologia di Attività Formativa:	**B - caratterizzante**
Docenti:	**Marco BOTTA (Titolare)**

1. Prerequisiti e Propedeuticità:

Competenze attese in ingresso

Lo studente deve conoscere prima di seguire il corso i fondamenti della programmazione e dei sistemi operativi.

Eventuali corsi propedeutici

Sistemi Operativi, Programmazione I e II

2. Obiettivi formativi:

Il corso si propone di fornire agli studenti nozioni base sulle reti 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.

3. Risultati dell'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.

4. Modalità di verifica dell'apprendimento:

La verifica viene fatta sulle conoscenze acquisite durante il corso. La durata della prova scritta è di 2 ore e prevede domande relative agli argomenti trattati nel Corso. L'esame può avere un seguito in forma orale, che è facoltativo per lo studente.

5. Modalità d'insegnamento:

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

6. Attività di supporto:

Diverso materiale reperibile tramite: Reti e Sistemi Distribuiti

7. Programma:

Fondamenti su reti di calcolatori.

Parte introduttiva alle reti (i livello di comunicazione

fisico, il livello link, reti wireless e mobilità) Il livello rete: instradamento e reti IP Il livello transport: controllo della congestione e del flusso endto-end Il livello applicativo: suite di applicazioni TCP/IP (Web, Posta, DNS, File transfer, sistemi P2P).

8. Testi consigliati e bibliografia:

J. Kurose - K. Ross. Computer Networking: A Top-Down approach featuring the Internet. 6/e
Pearson - Addison Wesley

Edizione italiana: J. Kurose - K. Ross. Reti di calcolatori e internet - Un approccio top-down 6a Edizione Pearson - Addison Wesley

Testo supplementari: D. Comer. Internetworking with TCP/IP Volume Unico, Prentice Hall

Insegnamento	**INF0002 - Servizi Web**

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

1. Prerequisiti e Propedeuticità:

Competenze attese in ingresso

Si richiede una buona conoscenza delle basi di dati (fornita dall'insegnamento Basi di Dati), dei sistemi operativi (fornita dall'insegnamento Sistemi Operativi), dell'analisi e della progettazione object-oriented (fornita dagli insegnamenti Programmazione II, e Algoritmi e strutture dati) e dei fondamenti della

programmazione distribuita (fornita dall'insegnamento Programmazione III). Gli studenti e le studentesse devono inoltre avere la capacità di scrivere, compilare e verificare la correttezza di programmi in Java.

Eventuali corsi propedeutici

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

2. Obiettivi formativi:

Questo insegnamento si colloca nell'area di approfondimento del percorso formativo del Corso di

Laurea in Informatica e ha come obiettivi formativi specifici l'insegnamento dei concetti fondamentali e delle metodologie per le applicazioni web e mobile.

L'implementazione delle applicazioni web richiede di guardare "dietro all'interfaccia utente" per andare a fondo su aspetti architetturali e tecnologici che possono influenzare non solo le prestazioni, la scalabilità e la robustezza, ma anche le tipologie di servizio che

possono essere effettivamente offerte.

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

Le tecnologie presentate sono note come Server-side Programming e riguardano la progettazione e lo sviluppo di applicazioni basate su architetture modulari che possono accedere a sorgenti dati eterogenee (come basi dati relazionali, file, etc.) allo scopo di fornire servizi complessi. Più precisamente, l'insegnamento tratterà dal punto di vista sia teorico che pratico: (i) lo sviluppo di pagine web statiche e dinamiche responsive; (ii) la programmazione lato server. Inoltre, l'insegnamento tratterà la rappresentazione e interpretazione dei dati in XML, data la sua importanza nella gestione dell'interoperabilità tra applicazioni.

In questo insegnamento, le capacita' di applicare le conoscenze acquisite vengono sviluppate nell'ambito di una sostanziale parte di laboratorio che permette di sperimentare le nozioni apprese durante le ore di teoria in aula, organizzando il lavoro in piccoli gruppi. I temi introdotti durante il laboratorio corredano e integrano le conoscenze derivanti dalla parte teorica (knowledge and understanding) e permettono agli studenti e alle studentesse di familiarizzare con le metodologie e tecnologie introdotte, anche investigando soluzioni alternative (applying knowledge and

understanding). Inoltre durante le ore di laboratorio è previsto lo sviluppo di un'applicazione realistica con interfaccia web che va oltre la complessita' del tipico esercizio svolto durante le ore di lezione. Lo svolgimento del progetto di laboratorio permette di allenare l'autonomia di giuzio nella scelta delle implentazioni da realizzare, analizzando in forma critica le informazioni acquisite nella parte teorica dell'insegnamento. La preparazione del progetto di laboratorio e' anche volta a stimolare la capacità comunicativa degli studenti

e delle studentesse che sono invitati/e a illustrare verbalmente le soluzioni adottate (communication skills).

3. Risultati dell'apprendimento attesi:

Gli studenti e le studentesse acquisiranno la conoscenza delle architetture di riferimento per lo sviluppo di applicazioni web modulari. Inoltre acquisiranno la conoscenza dei modelli più usati per gestire applicazioni interattive di media complessità, e delle tecnologie attualmente utilizzate per l'implementazione delle applicazioni e della loro interfaccia utente. Gli studenti e le studentesse acquisiranno la capacità di sviluppare pagine dinamiche PHP, e applicazioni Web a 3 livelli in ambiente Java, utilizzando HTML5-JavaScript, CSS, JSP e Java Servlet, e framework per applicazioni web MVC.

4. Modalità di verifica dell'apprendimento:

L'esame è composto da un TEST SCRITTO e da una VERIFICA DI LABORATORIO. Il primo e' volto a verificare che gli studenti e le studentesse abbiano acquisito la conoscenza fornita dall'insegnamento in termini di sapere. Il secondo e' volto a verificare le competenze acquisite (saper fare), nonche' le abilita' comunicative e l'autonomia di giudizio. Le due prove possono essere sostenute in qualsiasi ordine.

TEST SCRITTO

Prova scritta che include esercizi e domande teoriche sul programma dell'insegnamento. Viene valutata da un minimo di 0 ad un massimo di 30 e si considerano sufficienti i voti >=18. Come da regolamento di Ateneo, ogni studente/studentessa può sostenere un numero massimo di 3 prove scritte durante l'Anno Accademico (cioè, consegnare il proprio elaborato 3 volte). Il voto ottenuto durante un test scritto decade se si partecipa ad un altro test scritto e si consegna l'elaborato.

VERIFICA DI LABORATORIO

Prevede la discussione del progetto di laboratorio svolto

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

CALCOLO DEL VOTO FINALE DI ESAME

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

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

5. Modalità d'insegnamento:

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

La parte di teoria si svolge in lezioni frontali che

seguono il programma dell'insegnamento, integrate da casi di studio e da esercitazioni volte ad illustrare l'applicazione pratica dei concetti appena studiati.

La parte di laboratorio ha per oggetto l'insegnamento e la sperimentazione di tecnologie per lo sviluppo di pagine web statiche e dinamiche, e di applicazioni server-side MVC con accesso a basi di dati relazionali. Le esercitazioni propongono vari esercizi mirati a fornire esempi pratici, e sono svolte a calcolatore, utilizzando ambienti di sviluppo professionali, come Intellij IDEA e/o NetBeans IDE.

Le sperimentazioni che vengono effettuate durante le ore di laboratorio, strutturate come sequenze di esercizi specifici, sono fondamentali per aiutare gli studenti e le studentesse a comprendere e assimilare i contenuti teorici spiegati a lezione in quanto permettono di mettere in pratica i concetti e le metodologie illustrate su esempi concreti. Inoltre lo sviluppo del progetto di laboratorio, che si effettua in gruppo (max 3 persone per gruppo), permette di

consolidare le conoscenze teoriche e il lavoro in team in un caso realistico di media complessità. Si consiglia caldamente la frequenza costante alle lezioni di teoria e di laboratorio. È inoltre fondamentale iscriversi al corso online su I-Learn, all'interno del quale e' messo a disposizione materiale didattico di supporto.

6. Attività di supporto:

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

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

7. Programma:

Architetture delle applicazioni Web: Web browser e Web server.

Il Pattern Model View Controller per le applicazioni Web - Progettazione e sviluppo di applicazioni Web a 3 livelli basate su MVC:

- Il primo livello (client dell'applicazione): HTML5,
- CSS, scripting lato client (JavaScript e AJAX), Java Server Pages (JSP). Raccolta dati con HTML form.

Il secondo livello (logica applicativa): Servlet Java.

Il terzo livello (livello dei dati): accesso a database relazionali con Java Database Connectivity (JDBC), rappresentazione e gestione di informazioni con XML (XML Schema, XPath, XML Parsers).

- Framework le applicazioni web MVC.

Introduzione al linguaggio PHP per lo sviluppo di pagine web dinamiche.

8. Testi consigliati e bibliografia:

Libro di testo: Programmazione Web lato Server, di V. Della Mea, L. Di Gaspero, I. Scagnetto, Edizione 2, Apogeo, 2011

Altri testi (per gli interessati):

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

Insegnamento	**MFN0636 - Sicurezza**

Insegnamento (inglese):	**Computer and Network Security**
CFU:	**6**
Settore:	**INF/01 - INFORMATICA**
Periodo didattico:	**2**
Tipologia di Attività Formativa:	**B - caratterizzante** **D - libera**
Docenti:	**Francesco BERGADANO (Titolare)**

1. Prerequisiti e Propedeuticità:

Competenze attese in ingresso

Si presuppone la conoscenza dei sistemi operativi e delle reti di calcolatori basate sui protocolli della suite TCP/IP. La conoscenza di specifiche piattaforme di sviluppo e programmazione, quali Java e C++, considerata di aiuto alla comprensione degli argomenti svolti nel corso.

Eventuali corsi propedeutici

Sistemi Operativi, Reti di Calcolatori

2. Obiettivi formativi:

Il corso si propone di fornire agli studenti gli strumenti crittografici e tecnici utilizzati per garantire la sicurezza di reti e calcolatori. Inoltre, attraverso l'uso di esempi pratici, il corso fornisce agli studenti una comprensione concreta dei maggiori rischi di sicurezza e delle soluzioni disponibili

3. Risultati dell'apprendimento attesi:

Il corso si propone di preparare gli studenti a lavorare in azienda per la gestione dei sistemi informatici, cooperando con i livelli organizzativi per garantire la sicurezza del sistema informativo nel suo insieme.

4. Modalità di verifica dell'apprendimento:

La verifica viene fatta sulle conoscenze acquisite

durante il corso. La durata della prova scritta e' di un'ora e prevede domande aperte e domande a risposta multipla, relative agli argomenti trattati nel corso

5. Modalità d'insegnamento:

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

6. Attività di supporto:

slide online per alcune parti del corso, per il resto si fa riferimento al libro di testo.

7. Programma:

Strumenti crittografici: cifrari simmetrici e asimmetrici, funzioni di hash, firma elettronica Sicurezza della rete privata: analisi dei rischi di sicurezza informativa, controllo di accesso, protezione da virus, sistemi firewall, reti private virtuali

8. Testi consigliati e bibliografia:

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

Edizione, 1998

Altri Testi per consultazione: Bruce Schneier: Applied Cryptography, John Wiley and sons, 1994.
David Curry: Unix System Security, Addison-Wesley, 1992.

Insegnamento	**MFN0618 - Sistemi Informativi**

Insegnamento (inglese):	**Information Systems**
CFU:	**6**
Settore:	**INF/01 - INFORMATICA**
Periodo didattico:	**1**
Tipologia di Attività Formativa:	**D - libera**
Docenti:	**Roberto MICALIZIO (Titolare)**

1. Prerequisiti e Propedeuticità:

Competenze attese in ingresso

Si richiede allo studente di padroneggiare i concetti relativi alle basi di dati relazionali e alla modellazione concettuale (Entity-Relationship), alla programmazione e allo studio degli algoritmi.

Eventuali corsi propedeutici

Basi Dati e Sperimentazioni.

2. Obiettivi formativi:

L'insegnamento 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 buona parte dei processi aziendali sono ormai automatizzati. A tal fine, il corso introdurrà nozioni basilari di Business Process Management (BPM) e consentirà di sperimentare la modellazione di processi di business mediante lo standard Business Process Modeling Notation (BPMN).
Inoltre, l'insegnamento introduce lo studente ai sistemi di pianificazione delle risorse aziendali (Enterprise Resource Planning - ERP), ai sistemi integrati di gestione e profilazione del cliente (CRM), ai sistemi di supporto alla decisione (DSS) e alle piattaforme di Business Intelligence. In particolare, l'insegnamento si focalizzerà sulle metodologie di progettazione delle Data Warehouse e le metodologie di analisi on-line dei

dati (OLAP).

3. Risultati dell'apprendimento attesi:

Conoscenza delle principali ambiti di applicazione dei sistemi informativi. Capacità di progettazione di processi di business supportati dai sistemi informativi. Progettazione ed interrogazione delle Data Warehouse.

4. Modalità di verifica dell'apprendimento:

L'esame è composto da due parti. La prima parte consiste in un progetto di laboratorio in cui lo studente progetta un sistema informativo, un processo aziendale da esso supportato, e una data warehouse correlata. La seconda parte è una prova scritta in cui saranno verificate con domande aperte ed esercizi le competenze acquisite dallo studente.

5. Modalità d'insegnamento:

Le lezioni si svolgono in aula con l'ausilio di slides e di materiale distribuito via Moodle. Le esercitazioni si volgeranno in laboratorio tramite una suite di strumenti software open source (come Bonita BPM, PostgreSQL, Microsoft SQL Server Analysis Services.

6. Attività di supporto:

Si veda il sito Moodle del corso a cura del docente.

7. Programma:

1. Introduzione ai sistemi informativi aziendali (Modello organizzativo, modello funzionale, modelloinformatico)
2. Sistemi ERP (Le suite ERP; Paradigma ERP; Piattaforme software; Offerta ERP; Trasformazionedell'impresa)
3. Integrazione con il cliente: i sistemi CRM (Ruolo dei sistemi CRM nelle aziende; Schemaarchitetturale; Il paradigma; Esempi; Suite di package software;

Evoluzione dei CRM)

4. Piattaforme di Business Intelligence e DSS (Data Warehouse): Livello delle fonti ed ETL; Progettazione del sistema di warehousing; Livello di elaborazione: reporting e DSS; cenni di data mining

5. Gestione dell'azienda orientata ai processi (Studio e sperimentazione di diversi linguaggi dimodellazione quali: UML,e BPMN)

8. Testi consigliati e bibliografia:

- G. Bracchi, C. Francalanci, G. Motta, Sistemi informativi d'impresa, McGraw-Hill, Milano, 2010, ISBN: 978-88-386-6328-4
- M. Golfarelli, S. Rizzi, Data Warehouse - Teoria e pratica della progettazione 2/ed, McGrawHill, 2006, ISBN: 9788838662911

Libro di consultazione per la parte di Data Warehouse:

- Ralph Kimball, Margy Ross, The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling, John Wiley & Sons, 2nd

Edition, 2002

Libro di consultazione per la parte di Business Process Management:

- Antonio Di Leva, La Gestione dell'Azienda Basata sui Processi e i Sistemi Informativi Aziendali, Celid, 1st Edition, 2014, ISBN 978-88-6789-017-0

Insegnamento	**MFN0607 - Sistemi Intelligenti**

Insegnamento (inglese):	**Intelligent Systems**
CFU:	**6**
Settore:	**INF/01 - INFORMATICA**
Periodo didattico:	**2**
Tipologia di Attività Formativa:	**B - caratterizzante** **D - libera**
Docenti:	**Cristina BAROGLIO (Titolare)**

1. Prerequisiti e Propedeuticità:

Competenze attese in ingresso

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

Eventuali corsi propedeutici

Gli studenti che sono iscritti al corso di laurea di Informatica di Torino acquisiscono le competenze in ingresso sopra elencate seguendo gli insegnamenti di: - "Algoritmi e strutture dati", - "Programmazione I e II", - "Basi di dati", - "Matematica Discreta e Logica" (e sostenendo i relativi esami)

2. Obiettivi formativi:

Il corso si propone di fornire una introduzione generale alle problematiche nel settore dell'Intelligenza Artificiale, con particolare attenzione a come sia possibile costruire un sistema dotato di capacità autonome di risoluzione di problemi, di ragionamento e di apprendimento quando abbia a disposizione una rappresentazione simbolica del mondo.

Il corso si articola in tre parti principali: - Risoluzione automatica di problemi - Rappresentazione della conoscenza e ragionamento - Nozione di agente intelligente che agisce, ragiona ed apprende

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

3. Risultati dell'apprendimento attesi:

ITALIANO

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

Avranno inoltre acquisito le nozioni di base su architettura di un agente intelligente e su apprendimento automatico di conoscenza a partire da esempi.

Gli studenti acquisteranno una prima forma di consapevolezza sulla necessità di fare dei trade-off (in termini di qualità della soluzione e costo computazionale). tra strategie e metodologie alternative che risolvono la stessa classe di problemi.

L'esame orale richiede inoltre allo studente di acquisire una buona (discreta) capacità espositiva unita all'uso corretto della terminologia del settore (e più in generale dell'informatica).

ENGLISH

For what concerns the acquisition of methodological skills, students will learn the main search strategies for problem solving, basic knowledge about knowledge representation through logic formalisms and structered approaches (ontologies) and also the corresponding inference mechanisms.

They will also learn basic notions on agent architectures and on supervised machine learning.

Students will gain awareness of the need of a trade-off (in terms of solution quality and computational cost) between alternative strategies and methodologies for solving a same class of problems.

The oral examination requires also that students acquire good presentation skills and that they acquire competence in the correct use of the specialized vocabulary of the sector as well as of computer science in general.

4. Modalità di verifica dell'apprendimento:

ITALIANO

L'esame consiste in una interrogazione individuale sugli argomenti trattati nel corso. Tipicamente una delle domande richiede la capacità di applicare una metodologia/strategia per risolvere un semplice problema simile a quelli illustrati a lezione. Pertanto nell'esame orale, oltre a verificare la conoscenza dei concetti e delle metodologie, si intende verificare la capacità di applicazione dei concetti e metodologie a problemi specifici. Essendo l'esame un orale, è possibile inoltre verificare la chiarezza di esposizione e l'uso corretto della terminologia.

ENGLISH

The exam consists of an individual interview on the course topics. Typically one of the questions concerns the application of one of the methodologies/approaches studied during the course on a problem similar to one of those studied. Thus, the exam does not only verify the conceptual knowledge students have built but also the ability of practically using that knowledge. Moreover, the nature of the examination allows verifying

presentation clarity and the correct use of terminology.

5. Modalità d'insegnamento:

ITALIANO

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

ENGLISH

Lectures take place in a classroom. They concern both explanation of methodological aspects as well as exercises, solved collaboratively under the lead of the

teacher, and discussions of many examples. Some relevant tools are shown and their basic functioning is explained; some more complex tools that are/were used for tackling real world problems are also explained. Ethical concerns are also underlined.

6. Attività di supporto:

ITALIANO

Verra' reso disponibile sulla apposita pagina del corso all'interno del servizio I LEARN. Tale materiale prevede ulteriori esempi di utilizzo dei meccanismi di ragionamento e di strategie di ricerca oltre a quelli contenuti sul libro di testo.

ENGLISH

Additional material is made available through the platform I LEARN. This includes further examples of application of reasoning and search strategies.

7. Programma:

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

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

Parte 2) RAPPRESENTAZIONE DELLA CONOSCENZA E

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

Parte 3) AGENTI E APPRENDIMENTO AUTOMATICO In questa parte conclusiva si introduce la nozione di agente

intelligente che opera in un ambiente e si fa vedere come l'agente possa avere sia comportamenti reattivi che deliberativi a seconda del compito assegnato. Si illustra come agente debba avere capacità di risoluzione automatica di problemi e di ragionamento sullo stato del mondo e sul suo stato. Si descrive brevemente come l'apprendimento automatico sia una delle caratteristiche essenziali per ottenere un agente intelligente. Vengono introdotte solo nozioni elementari con particolare riguardo all'apprendimento da esempi (in particolare apprendimento di alberi di decisione). Viene infine fatta una introduzione alle reti neuronali come strumento per passare dal livello sub simbolico a quello simbolico.

8. Testi consigliati e bibliografia:

S. J. Russell e P. Norvig, Artificial Intelligence: a modern approach , Third edition, 2010, Pearson
Nell'ambito del corso verranno affrontate le problematiche descritte nei capitoli 1,2,3,5,6,7,8,9, 18 del testo. Si ricorda che esiste anche la traduzione italiana della prima parte della terza edizione (volume 1) e la traduzione integrale della seconda edizione

(volume 1 e 2), entrambe edite da Pearson/Prentice Hall.

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

Insegnamento	**MFN0601 - Sistemi Operativi**

Insegnamento (inglese):	**Operating Systems**
CFU:	**12**
Settore:	**INF/01 - INFORMATICA**
Periodo didattico:	**1**
Tipologia di Attività Formativa:	**B - caratterizzante**
Docenti:	**Cristina BAROGLIO (Titolare)** **Enrico BINI (Titolare)** **Massimiliano DE PIERRO (Titolare)** **Daniele GUNETTI (Titolare)** **Daniele Paolo RADICIONI (Titolare)** **Claudio SCHIFANELLA (Titolare)**

1. Prerequisiti e Propedeuticità:

Competenze attese in ingresso

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

Eventuali corsi propedeutici

Costituiscono prerequisiti i contenuti dei corsi di:

* Architettura degli Elaboratori I * Programmazione I

2. Obiettivi formativi:

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

Per la parte di laboratorio gli obiettivi formativi sono l'apprendimento del linguaggio C, utilizzato per la programmazione nell'ambiente del sistema operativo Unix. La parte di laboratorio mira a fornire allo studente una conoscenza (teorica e pratica) di base sui

comandi della shell, sulla gestione dei processi, sugli strumenti di inter-process communication e sulla gestione dei segnali forniti dal sistema, oltre che alcuni rudimenti di programmazione bash.

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

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

Sistemi Operativi

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

3. Risultati dell'apprendimento attesi:

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

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

4. Modalità di verifica dell'apprendimento:

Per superare l'esame di sistemi operativi lo studente deve ottenere un punteggio sufficiente (almeno 18) nelle prove in cui si articola l'esame, che dovranno essere sostenute nel seguente ordine:

-Prova di laboratorio: consiste nella discussione individuale del progetto di laboratorio ed èfinalizzata a valutare le competenze acquisite nei moduli di laboratorio. Più precisamente saranno valutati sia gli strumenti insegnati nel modulo Unix, sia le competenze inerenti il linguaggio C.

-Prova scritta: è finalizzata a valutare le competenze acquisite sugli argomenti insegnati nelmodulo di teoria e C. In generale potrà essere composta da domande aperte, domande chiuse a scelta multipla, esercizi.

-Orale facoltativo. Può essere sostenuto solo avendo preso nello scritto almeno 26/30. Integra ilvoto ottenuto sostenendo le parti di esame 1) e 2), e permette di prendere un voto massimo di 30/30 e lode.

Il voto finale sarà ottenuto come somma pesata dei voti

conseguiti nelle prove descritte.

5. Modalità d'insegnamento:

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

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

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

esercizi miranti a fornire esempi pratici.

6. Attività di supporto:

Dispense e lucidi usati a lezione, programmi di esempio.

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

qui

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

Sono inoltre previste 12 ore facoltative di esercitazione e in cui saranno proposti esercizi e relative soluzioni sugli argomenti principali del corso e saranno revisionati gli aspetti teorici relativi.

7. Programma:

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

PARTE DI TEORIA:

* Introduzione al Corso di Sistemi Operativi

* PARTE I: GENERALITA'o Introduzione (cap. 1) o Strutture dei Sistemi Operativi (cap. 2) * PARTE II: GESTIONE DEI PROCESSI o Processi (cap. 3) o Thread (cap. 4) o Scheduling della CPU (cap. 5) o Sincronizzazione dei Processi (cap. 6) o Deadlock (Stallo di Processi) (cap. 7)
* PARTE III: GESTIONE DELLA MEMORIA (PRIMARIA)o Memoria Centrale (cap. 8) o Memoria Virtuale (cap. 9)
* PARTE IV: GESTIONE DELLA MEMORIA SECONDARIA

o Interfaccia del File System (cap. 10) o Realizzazione del File System (cap. 11) o Memoria Secondaria e Terziaria (Gestione

dell'Hard disk)

= =

PARTE DI LABORATORIO:

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

System call per la creazione e la sincronizzazione di processi

System call per L'InterProcess Communication e per la gestione di segnali

Esercitazioni pratiche, in particolare: esercitazioni finalizzate ad imparare il linguaggio C, ad utilizzare Unix e a sviluppare programmi concorrenti

8. Testi consigliati e bibliografia:

Teoria (TESTO DI RIFERIMENTO):

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

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

Insegnamento	INF0004 - Storia dell'Informatica

Insegnamento (inglese):	History of Computer Science
CFU:	6
Settore:	INF/01 - INFORMATICA
Periodo didattico:	1 2
Tipologia di Attività Formativa:	D - libera
Docenti:	Felice CARDONE (Titolare) Daniele GUNETTI (Titolare)

1. Prerequisiti e Propedeuticità:

Competenze attese in ingresso

Conoscenza di base dell'architettura di un computer e di un sistema operativo. Dimestichezza di base con i principali paradigmi di programmazione.

Eventuali corsi propedeutici

Architetture degli elaboratori I. Programmazione I e II. Sistemi Operativi.

2. Obiettivi formativi:

Dare una visione di insieme e una prospettiva storica dei momenti salienti dell'evoluzione delle idee teoriche e pratiche dell'informatica.

3. Risultati dell'apprendimento attesi:

Acquisire una visione di insieme e una prospettiva storica dei momenti salienti dell'evoluzione delle idee teoriche e pratiche dell'informatica.

4. Modalità di verifica dell'apprendimento:

Esame scritto

5. Modalità d'insegnamento:

Lezioni frontali in aula

6. Attività di supporto:

7. Programma:

Il corso fornisce una introduzione alla storia dell'informatica, con 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.

La prima parte del corso sarà dedicata ad 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.

La seconda parte comprende, accanto ad un'introduzione istituzionale alla storia del calcolo basata su uno dei libri di testo elencati, anche una serie di lezioni monografiche dedicate a temi e lavori classici in varie aree dell'informatica.
L'esame del corso prevede un compito finale il cui scopo sarà quello di verificare la comprensione della portata tecnologica e culturale degli eventi salienti della storia dell'informatica.
AVVISO per gli studenti del DAMS che hanno in piano di studi l'insegnamento di Fondamenti di Informatica II, che quest'anno mutua da Storia dell'informatica. Si intende che la parte mutuata è la seconda, che sarà tenuta nel secondo semestre dal Prof. Cardone.

8. Testi consigliati e bibliografia:

M. Campbell-Kelly and W. Aspray. Computer: A History of the Information Machine. Basic Books, 1996.

P.E. Ceruzzi. A History of Modern Computing (2nd edition ed.). MIT Press, 2003.

M. Davis. Il calcolatore universale. Da Leibniz a Turing. Adelphi, 2012.

M. Priestley. A Science of Operations: Machines, Logic and the Invention of Programming. Springer, 2011.

B. Randell (Ed.), The Origins of Digital Computers: Selected Papers (3rd ed.). Springer-Verlag, 1982.

Insegnamento	**MFN0606 - Sviluppo delle Applicazioni Software**

Insegnamento (inglese):	**Development of Software Applications**
CFU:	**9**
Settore:	**INF/01 - INFORMATICA**
Periodo didattico:	**2**
Tipologia di Attività Formativa:	**B - caratterizzante**
Docenti:	**Matteo BALDONI (Titolare)** **Sara CAPECCHI (Titolare)** **Claudia PICARDI (Titolare)**

1. Prerequisiti e Propedeuticità:

Competenze attese in ingresso

Il corso assume le seguenti conoscenze:
- familiarità con le tecniche di programmazione in linguaggi di alto livello come Java; -

conoscenza dei contetti di base di progettazione e interrogazione di basi di dati.

Eventuali corsi propedeutici

Le competenze del primo punto sono fornite dai corsi di programmazione della laurea triennale, come i corsi di Programmazione I e Programmazione II. Le competenze del secondo punto da un corso di introduzione alle basi di dati, come il corso di Basi di Dati.

2. Obiettivi formativi:

L'insegnamento concorre al raggiungimento degli obiettivi formativi del Corso di Laurea in Informatica fornendo i concetti di base dell'ingegneria del software e allo sviluppo di applicazioni software, utilizzando la metodologia Agile Unified Process (UP), che sfrutta il linguaggio di modellazione UML. Si propone perciò di fornire le competenze teoriche e pratiche necessarie per sviluppare un'applicazione significativa individuando con chiarezza la logica applicativa, l'interazione con le basi di dati e le interfacce richieste dai requisiti. Inoltre dovrà imparare a pianificare il lavoro secondo i canoni

dello sviluppo dei progetti: lavoro di gruppo, definizione degli obiettivi e delle fasi di sviluppo.

3. Risultati dell'apprendimento attesi:

Conoscenza e capacità di comprensione

Al termine dell'insegnamento lo studente dovrà conoscere:
- i concetti di base dell'ingegneria del software;
- la metodologia UP e le modalità di utilizzo del linguaggio UML in tale ambito;- i design pattern di progettazione e architetturali.

Capacità di applicare conoscenza e comprensione

Al termine dell'insegnamento lo studente dovrà essere in grado di:
- applicare UP alla progettazione di un applicativo;
- individuare l'utilità e l'applicabilità dei design pattern al progetto di un software.

Autonomia di giudizio

Alla fine di questo insegnamento lo studente saprà:

- sviluppare un'applicazione significativa individuando con chiarezza la logica applicativa;
- l'interazione con le basi di dati e le interfacce richieste dai requisiti;

- pianificare il lavoro secondo i canoni dello sviluppo dei progetti: lavoro di gruppo, definizionedegli obiettivi e delle fasi di sviluppo.

Abilità comunicative

Al termine dell'insegnamento lo studente dovrà essere in grado di:

- esporre, in forma scritta e orale, una relazione tecnica che spieghi e motivi le scelte progettualinel caso di studio proposto.

4. Modalità di verifica dell'apprendimento:

L'esame consiste in una parte di teoria e una parte di progetto.

La parte di progetto viene svolta in gruppo in parte durante il laboratorio e in parte nel proprio tempo di

studio individuale. La valutazione del progetto avviene tramite una discussione di gruppo.

La parte di teoria consiste di una prova scritta.

5. Modalità d'insegnamento:

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

Per la parte di teoria, le lezioni si svolgono con il supporto di slide e lavagna.

La parte di laboratorio, consiste nello sviluppo guidato di un progetto, realizzato in gruppi utilizzando la metodologia UP.

6. Attività di supporto:

La documentazione relativa al laboratorio e le slide del corso saranno disponibili sul supporto online al corso I-learn.

7. Programma:

1) Elementi di ingegneria del software: modelli Waterfall, Spirale, V-shaped, Componend-basedDevelopment, metodologie Agili.

2) Introduzione all'UML: use case diagram, class diagram, object diagram, sequence diagram,communication diagram, state chart, activity diagram.

3) Una metodologia Agile: Unified Process (UP). Fasi di sviluppo (ideazione, elaborazione,costruzione). Casi d'uso, modello di dominio, diagrammi di Sequenza di Sistema, Contratti, Architettura logica e organizzazione in layer, Progettazione (DSD e DCD), GRASP, GoF.

4) Testing: unit testing, acceptance test, white e black box testing.

8. Testi consigliati e bibliografia:

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

Altri testi consigliati:

Ian Sommerville. Software Engineering, 10/e. Pearson.

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

Insegnamento	**MFN0634 - Tecnologie Web**

Insegnamento (inglese):	**Web Technologies**
CFU:	**6**
Settore:	**INF/01 - INFORMATICA**
Periodo didattico:	**1**
Tipologia di Attività Formativa:	**B - caratterizzante** **D - libera**
Docenti:	**Marco BOTTA (Titolare)** **Giancarlo Francesco RUFFO (Titolare)**

1. Prerequisiti e Propedeuticità:

Competenze attese in ingresso

Lo studente deve mostrare di possedere una buona familiarità con i principi della progammazione (imperativa, ad oggetti e basata su eventi). Inoltre, deve conoscere le basi operative per gestire una base di dati basata su SQL e per configurare/installare pacchetti software nel proprio sistema operativo.

Eventuali corsi propedeutici

- MFN0582 - Programmazione I
- MFN0585 - Programmazione II
- MFN0601 - Sistemi Operativi
- MFN0602 - Basi di Dati

2. Obiettivi formativi:

Gli obiettivi di questo corso sono i seguenti:

- Imparare a produrre siti Web dinamici, animati, interattivi e collegati ad un database in back end;
- Imparare diversi linguaggi e tecnologie per lo

sviluppo Web client-side, quali HTML5, CSS, JavaScript, JQuery

- Imparare principi della programmazione server side tramite principalmente PHP e MySQL, sfruttando strumenti opensource come i comuni browser web e il server web Apache.

3. Risultati dell'apprendimento attesi:

HTML5, PhP, JavaScript, CSS, Web services, mySQL, version control

4. Modalità di verifica dell'apprendimento:

Esercizi di laboratorio (almeno 4) (10%): saranno corretti e valutati i vari esercizi di laboratorio consegnati.

Relazione (35%): deve contenere la descrizione del progetto che si intende realizzare e deve essere consegnata almeno un mese prima della data di appello scelta (2000-3000 parole). Conterrà una descrizione funzionale, il wireframe del sito ed altre specifiche che saranno definite dal docente durante il corso.

Progetto finale (55%): progetto individuale che implementa quanto dichiarato dallo studente nella relazione consegnata precedentemente. Il sito sarà sottoposto alle opportune fasi di test e valutato di conseguenza.
Gli studenti non frequentanti che non potranno consegnare gli esercizi di laboratorio, potranno sostenere un esame orale sostitutivo basato sul progetto consegnato.

5. Modalità d'insegnamento:

Le lezioni teoriche saranno svolte in modalità tradizionale/frontale, con l'ausilio di diapositive che saranno proiettate in aula. La parte applicativa del corso sarà svolta in laboratorio informatico e gli studenti saranno incentivati a svolgere gli esercizi in aula e consegnarli per la correzione. Le diapositive saranno messe a disposizione degli studenti come materiale integrativo.

6. Attività di supporto:

Il materiale didattico di supporto sarà reso tramite un'apposita pagina web accessibile tramite il sistema di supporto alla didattica Moodle gestito dal nostro Corso di Laurea.

7. Programma:

- Progettazione base ed implementazione di siti Web
- Presentazione delle diverse strategie di navigazione e di organizzazione dei siti
- Tecnologie client-side, tra cui HTML5, CSS, Javascript, JSON e JQuery

Tecnologie server side, facendo particolare attenzione alle implementazioni in PhP

Gestione dei dati in back end

Tecnologie emergenti (MVC, bootstrap, angular, versioning con github, etc.)

8. Testi consigliati e bibliografia:

J. Miller, V. Kirst, Marty Stepp. [Web Programming Step by Step

http://www.webstepbook.com/index.sh... . 2nd edition (2012)

Powered by TCPDF (www.tcpdf.org)