



UNIVERSITÀ
DEGLI STUDI
DI PADOVA

SCUOLA DI SCIENZE

Bollettino Notiziario

Anno Accademico 2018/2019

Laurea in Informatica (Ord. 2011)

Curriculum: Corsi comuni

ALGEBRA E MATEMATICA DISCRETA

(Titolare: Prof.ssa GEMMA PARMEGGIANI)

Periodo: I anno, 2 semestre
Indirizzo formativo: Corsi comuni
Tipologie didattiche: 58A+40E; 12,00 CFU

Prerequisiti :

Abilità analitiche (abilità di ragionamento logico); conoscenze e abilità come specificato nel Syllabus della pagina del Corso di laurea in informatica. In particolare:

- strutture numeriche (numeri naturali, numeri primi, frazioni numeriche, numeri razionali, elementi dei numeri reali, disuguaglianze, valore assoluto, potenze, radici);
- algebra elementare (calcolo letterale, polinomi e operazioni fra polinomi, identità, equazioni di primo e secondo grado, sistemi lineari);
- insiemi e funzioni (linguaggi degli insiemi, nozione di funzione, grafici di funzioni notevoli, concetto di condizione sufficiente, necessaria);
- geometria (geometria euclidea piana, angoli, radianti, aree e figure simili, nozione di luogo geometrico, proprietà dei triangoli, dei parallelogrammi, dei cerchi, simmetrie, similitudini e trasformazioni nel piano, coordinate cartesiane ed equazioni di semplici luoghi geometrici, elementi di trigonometria, elementi di geometria euclidea nello spazio, volumi).

Conoscenze e abilità da acquisire :

Obiettivo del corso è quello di permettere allo studente di sviluppare le proprie capacità analitiche e di acquisire alcune conoscenze di base riguardanti l'algebra, la geometria e la matematica discreta.

Attività di apprendimento previste e metodologie di insegnamento :

Lezioni frontali ed esercitazioni.

Contenuti :

Massimo comun divisore e algoritmo di Euclide; anelli di classi resto. Richiami sui polinomi: divisione, zeri, fattorizzazione in irriducibili (sui reali e sui complessi). Equazioni lineari e matrici: matrici, operazioni sulle matrici, sistemi di equazioni lineari, metodo di eliminazione di Gauss, sistemi omogenei, matrice inversa. Spazi vettoriali, sottospazi, basi. Funzioni lineari, nucleo e immagine. Autovalori, autovettori e diagonalizzazione di matrici. Prodotti scalari, ortogonalità e procedimento di Gram-Schmidt.

Teoria dei grafi: introduzione ai grafi e nozioni di base, connettività, cammini, tagli, alberi, grafi planari, cicli euleriani e circuiti hamiltoniani.

Enumerazione: permutazioni e combinazioni semplici, permutazioni e combinazioni con ripetizione, distribuzioni, identità binomiali e triangolo di Pascal, relazioni di ricorrenza.

Modalità di esame :

Esame scritto. Lo scritto prevede domande ed esercizi volti a valutare il livello di apprendimento delle nozioni impartite durante il corso e la capacità di elaborarle ed applicarle.

Criteri di valutazione :

I criteri per una valutazione positiva sono:

- la correttezza e la completezza delle soluzioni degli esercizi
- la proprietà del linguaggio matematico utilizzata

Testi di riferimento :

Alan Tucker, Applied Combinatorics. : Wiley and Sons, 2007

Marco Abate e Chiara de Fabritiis, Geometria analitica con elementi di algebra lineare. : McGraw-Hill,

Eventuali indicazioni sui materiali di studio :

Materiale didattico predisposto dai docenti e reso disponibile

- per la parte di Algebra sulla pagina web: http://www.math.unipd.it/~parmeggi/mat_gemma.html
- per la parte di Matematica Discreta sulla piattaforma Moodle.

ALGORITMI E STRUTTURE DATI

(Titolare: Prof. PAOLO BALDAN)

Periodo: Il anno, 2 semestre
Indirizzo formativo: Corsi comuni
Tipologie didattiche: 56A+16E; 9,00 CFU
Sede dell'insegnamento : Plesso Paolotti, Via Luzzatti.
Aule : LuM250

Prerequisiti :

Elementi di programmazione

Matematica discreta

Analisi matematica

Conoscenze e abilità da acquisire :

Il corso intende fornire un'introduzione agli algoritmi e alla loro analisi. Lo studente apprende alcuni algoritmi e strutture dati fondamentali che sono alla base dello sviluppo dei sistemi software. L'analisi di tali algoritmi aiuta lo studente a sviluppare una sensibilità per la realizzazione di programmi efficienti e corretti.

Attività di apprendimento previste e metodologie di insegnamento :

Lezioni ed esercitazioni.

Contenuti :

- Fondamenti

Analisi dettagliata di InsertSort: Pseudocodice. Calcolo diretto del tempo calcolo in funzione di n . Tasso di crescita e notazione asintotica. L'algoritmo MergeSort e la tecnica divide et impera. Analisi della complessità di MergeSort. Soluzione delle ricorrenze. Il teorema dell'esperto. QuickSort. Complessità media di QuickSort e analisi probabilistica. Randomizzazione di QuickSort.

- Ordinamento e Statistiche d'Ordine

HeapSort e sua analisi. Limite inferiore per la complessità degli algoritmi di ordinamento. Implementazione di code con priorità mediante heap. Ordinamento in tempo lineare. Algoritmi CountingSort e RadixSort.

- Strutture Dati

Tavole hash. Alberi binari di ricerca. Alberi rosso-neri. Aumento di strutture dati. Teorema dell'aumento per alberi rosso-neri. Alberi di intervalli.

- Tecniche avanzate di progettazione e analisi

Programmazione dinamica. Algoritmi golosi. Analisi ammortizzata.

Modalità di esame :

Prova scritta ed esame orale.

Criteri di valutazione :

Gli esercizi della prova scritta mirano a valutare la capacità dello studente di utilizzare le nozioni apprese per l'individuazione di soluzioni algoritmiche efficienti a problemi assegnati. La prova orale verifica la conoscenza degli argomenti teorici discussi a lezione.

Testi di riferimento :

T.H.Cormen, C.E.Leiserson, R.L.Rivest, C.Stein, *Introduzione agli Algoritmi e Strutture Dati* (terza edizione). : McGraw-Hill Italia, 2010

Eventuali indicazioni sui materiali di studio :

Esercizi e materiale aggiuntivo sono resi disponibili tramite la pagina web del corso e la pagina moodle.

ANALISI MATEMATICA

(Titolare: Dott.ssa CATERINA SARTORI)

Periodo: I anno, 1 semestre
Indirizzo formativo: Corsi comuni
Tipologie didattiche: 64A+32E; 12,00 CFU

Prerequisiti :

Matematica di base (disequazioni, coordinate cartesiane, funzioni trigonometriche, logaritmiche ed esponenziali).

Conoscenze e abilità da acquisire :

Il corso si propone di illustrare i concetti e gli strumenti dell'Analisi per funzioni di una variabile reale, dando particolare rilievo agli aspetti di base del calcolo integro-differenziale.

Attività di apprendimento previste e metodologie di insegnamento :

Durante il semestre sono previste delle prove parziali di accertamento che vengono corrette dall'insegnante. Se lo studente supera tutte le prove in modo sufficiente, il voto finale dell'esame si ottiene facendo la media dei voti delle singole prove.

Contenuti :

Numeri (naturali, interi, razionali, reali, complessi. Cardinalità: insiemi finiti e infiniti). Piano e Spazio euclidei (vettori nel piano e nello spazio ordinario; equazioni cartesiane di rette e piani). Successioni in \mathbb{R} . Limiti di funzioni di una variabile reale. Derivate di funzioni di una variabile reale. Teoremi fondamentali del calcolo integro-differenziale. Formula di Taylor. Massimi e minimi locali. Grafici di funzioni di una variabile. Integrale definito. Integrale indefinito e metodi di integrazione. Integrali generalizzati. Serie numeriche. Equazioni differenziali del primo ordine. Cenni su alcune generalizzazioni dell'Analisi per funzioni di π^1 variabili.

Modalità di esame :

L'esame consiste di due parti: la prima \tilde{A} dedicata agli esercizi, la seconda alla teoria alla base degli esercizi.

Le due parti possono essere superate durante uno stesso appello o in due consecutivi entro la stessa sessione.

Criteri di valutazione :

Comprensione degli argomenti teorici e capacità di risolvere esercizi.

In particolare allo studente viene richiesto

- 1) di usare il linguaggio matematico correttamente,
- 2) di dimostrare in modo rigoroso un certo numero di teoremi,
- 3) di sviluppare un approccio critico che gli permetta di individuare eventuali errori di un ragionamento matematico

Testi di riferimento :

LUCA BERGAMASCHI, *Fondamenti di Analisi Matematica 1.* : Ed. Libreria Progetto, via Marzolo 2, 2017

Eventuali indicazioni sui materiali di studio :

In classe saranno dati consigli per eventuali libri di testo.
Tutto il materiale didattico presentato nelle lezioni frontali \tilde{A} reso disponibile sulla piattaforma MOODLE.

ARCHITETTURA DEGLI ELABORATORI

(Titolare: Prof. ALESSANDRO SPERDUTI)

Periodo: I anno, 1 semestre
Indirizzo formativo: Corsi comuni
Tipologie didattiche: 40A+20E+10L; 8,00 CFU
Sede dell'insegnamento : Padova
Aule : LUM 250, Paolotti

Prerequisiti :

Non sono richieste conoscenze particolari, se non quelle di base della matematica.

L'insegnamento non prevede propedeuticit .

Conoscenze e abilita' da acquisire :

Obiettivo dell'insegnamento   quello di permettere allo studente di acquisire alcune conoscenze di base funzionali e tecnologiche riguardanti l'architettura degli elaboratori limitatamente al loro utilizzo in ambito locale. Sono previste esercitazioni in laboratorio informatico che consentono allo studente di approfondire le conoscenze acquisite mediante l'utilizzo di semplici simulatori di CPU, Cache, e Pipeline.

Attivit  di apprendimento previste e metodologie di insegnamento :

L'insegnamento prevede lezioni frontali ed esercitazioni in laboratorio informatico. Le esercitazioni in laboratorio informatico consistono nella sperimentazione da parte degli studenti di simulatori di CPU, Cache e Pipeline, sotto vari scenari operativi. In questo modo gli studenti possono verificare sperimentalmente i concetti appresi a lezione e acquisire sia capacit  di applicazione dei concetti appresi che di giudizio critico.

Contenuti :

La struttura e le tematiche dell'insegnamento saranno le seguenti:

- Introduzione:

Evoluzione dei calcolatori; visione ad alto livello della struttura di un calcolatore; struttura e funzione della Cpu.

- Gestione della Memoria:

Memorie e Gerarchie di Memorie. Cache: tecniche di associazione, politiche di rimpiazzo. Simulatore Cache.

- Dispositivi e Gestione dell'Input/Output:

Input/Output: dispositivi esterni, modulo I/O, gestione da programma, gestione tramite interruzioni, DMA.

- Cenni di Circuiti Combinatori e Sequenziali, Microprogrammazione:

Algebra di Boole. Porte logiche. Circuiti Combinatori. Circuiti sequenziali. Microprogrammazione.

- Aritmetica dei Calcolatori:

Livello Macchina, Rappresentazione Binaria, Aritmetica.

- Linguaggio Assembler e Livello Instruction Set:

Linguaggio assembler. Caratteristiche istruzioni macchina. Tipi degli operandi, dati, operazioni. Indirizzamento. Formato istruzioni.

Simulatore CPU.

- Livello Instruction Set:

Architetture CISC e RISC, Processori Multicore

- Valutazione e Miglioramento delle Prestazioni:

Pipeline: principi generali, prestazioni ideali, dipendenze, tecniche per la riduzione delle dipendenze, MIPS. Simulatore pipeline MIPS.

Modalit  di esame :

Lo studente deve superare un esame scritto e, se ritenuto necessario dal docente, un esame orale.

Criteri di valutazione :

Il testo dell'esame scritto contiene alcune domande che consentono di valutare il livello di apprendimento delle nozioni impartite durante l'insegnamento e la capacit  dello studente nell'analizzarle criticamente. Sono poi presenti esercizi in cui si richiede allo studente di ricostruire il funzionamento o il dimensionamento di alcune componenti dell'elaboratore. Tali esercizi hanno lo scopo di valutare se lo studente ha sviluppato la capacit  di applicare le nozioni apprese durante l'insegnamento.

Nel caso in cui la valutazione dello scritto risulti appena sotto la sufficienza, il docente pu  decidere di integrare l'esame scritto con un esame orale per meglio verificare la preparazione dello studente.

Testi di riferimento :

Stallings, William, Computer organization and architecture designing for performance William Stallings. Upper Saddle River: Pearson education, 2015

Eventuali indicazioni sui materiali di studio :

Vengono rese disponibili, come riferimento, i lucidi utilizzati a lezione.

AUTOMI E LINGUAGGI FORMALI

(Titolare: Prof. GILBERTO FILE')

Periodo: Il anno, 2 semestre

Indirizzo formativo: Corsi comuni

Tipologie didattiche: 48A+16E; 8,00 CFU

Sede dell'insegnamento : Plesso Paolotti, via Paolotti.

Aule : LUM 250.

Prerequisiti :

Nozioni di logica.

Conoscenze e abilita' da acquisire :

Questo corso fornisce i concetti fondamentali della teoria degli automi e dei linguaggi formali, mostrando la loro applicazione ai compilatori.

Inoltre, introduce le nozioni di indecidibilit  e intrattabilit .

Attivit  di apprendimento previste e metodologie di insegnamento :

Lezioni frontali ed esercizi in classe.

Contenuti :

I principali contenuti del corso sono i seguenti:

Parte 1: regular languages and lexical analysis (3 cfu)

-- automi a stati finiti

-- espressioni e linguaggi regolari

- pumping lemma
- propriet  dei linguaggi regolari
- analisi lessicale.

Parte 2: linguaggi liberi da contesto e analisi sintattica (3 cfu)

- grammatiche e linguaggi liberi da contesto
- automi a pila
- pumping lemma
- propriet  dei linguaggi liberi da contesto
- analisi sintattica.

Parte 3: indecidibilit  e intrattabilit  (2 cfu)

- macchine di Turing
- indecidibilit 
- problemi intrattabili
- classi P e NP.

Modalit  di esame :

Scritto

Criteri di valutazione :

Lo scritto contiene alcune domande che consentono di valutare il livello di apprendimento delle nozioni impartite durante il corso. Sono poi presenti esercizi di costruzione di automi e di grammatiche formali.

Testi di riferimento :

Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman, *Compilers: Principles, Techniques, and Tools.* : Addison-Wesley, 2006
Hopcroft, John E.; Motwani, Rajeev; Ullman, Jeffrey D., *Automi, linguaggi e calcolabilit * . Milano: Addison-Wesley Pearson Education Italia, 2009

Eventuali indicazioni sui materiali di studio :

Un sito moodle con tutto il materiale del corso. Sono disponibili anche e le registrazioni delle lezioni

BASI DI DATI

(Titolare: Prof. MAURO CONTI)

Periodo: Il anno, 1 semestre
Indirizzo formativo: Corsi comuni
Tipologie didattiche: 48A+8E+16L; 9,00 CFU

Prerequisiti :

  opportuno avere familiarit  con gli elementi di base della programmazione, cos  come forniti nel corso di "Programmazione". L'insegnamento non prevede propedeuticit .

Conoscenze e abilit  da acquisire :

Obiettivo del corso   quello di permettere allo studente di acquisire alcune conoscenze di base riguardanti le funzionalit  dei DBMS (Data Base Management Systems " sistemi per la gestione di basi di dati), la progettazione delle basi di dati mediante l'uso di modelli concettuali, il progetto logico mediante il modello relazionale dei dati e l'uso del linguaggio SQL per la definizione e la realizzazione di basi di dati. Sono previste esercitazioni in laboratorio che consentono allo studente di sperimentare le conoscenze acquisite mediante l'utilizzo di un DBMS concreto.

Infine   richiesto lo sviluppo di un progetto che consiste nella progettazione e realizzazione di una base di dati (analisi, progetto concettuale, progetto logico, implementazione, individuazione e implementazione di query di interesse). Oltre ad incrementare la capacit  di applicare le conoscenze teoriche acquisite, le varie fasi del progetto espongono lo studente a scelte autonome e ragionate, che richiedono una rielaborazione personale delle conoscenze acquisite.

Attivit  di apprendimento previste e metodologie di insegnamento :

L'insegnamento prevede lezioni frontali, esercitazioni in laboratorio e la realizzazione di un progetto. Le esercitazioni in laboratorio consistono nell'implementazione di semplici database, relative interrogazioni SQL. Sono propedeutiche alla realizzazione del progetto, nel quale lo studente individua autonomamente un caso di studio, progetta ed implementa una base di dati.

Contenuti :

La struttura e le tematiche del corso saranno le seguenti:

- Introduzione

Basi di dati e sistemi di gestione di basi di dati. Architettura di un sistema di gestione di basi di dati.

- La progettazione concettuale di una base di dati

Il modello concettuale a oggetti. Progettazione concettuale di basi di dati mediante il modello a oggetti.

- La progettazione logica di una base di dati

Il modello relazionale. L'algebra relazionale e il calcolo relazionale. Progettazione logica di basi di dati mediante traduzione di schemi a oggetti in schemi relazionali.

- La normalizzazione di schemi relazionali

Dipendenze funzionali. Scomposizione di relazioni, con preservazione di dati e/o dipendenze. Forme normali per gli schemi di relazione (1NF, 2NF, 3NF e BCNF) e decomposizioni in forma normale.

- Il linguaggio SQL

Il data definition language di SQL. Il data manipulation language di SQL (interrogazione e aggiornamento). Conoscenza procedurale: procedure e trigger. Transazioni. Controllo degli accessi (basato sui privilegi). Il DBMS MySQL.

- SQL per le applicazioni
SQL embedded. Call level interface (JDBC, ODBC).

Modalità di esame :

Lo studente deve superare uno scritto e realizzare un progetto (nel quale mettere in pratica le nozioni acquisite nel corso). Il progetto Ã" poi discusso in forma orale.

Criteri di valutazione :

Lo scritto contiene alcune domande che consentono di valutare il livello di apprendimento delle nozioni impartite durante il corso. Sono poi presenti esercizi di progettazione, formulazione di query SQL e normalizzazione che richiedono allo studente un'elaborazione personale di concetti e tecniche viste nel corso.

La valutazione del progetto considera la capacitÃ , da parte dello studente, di individuare un caso di studio adeguato, di svolgere in modo autonomo un'attivitÃ di progettazione qualitativamente appropriata e di realizzare una implementazione disciplinata.

Testi di riferimento :

Paolo Atzeni, Stefano Ceri, Stefano Paraboschi, Riccardo Torlone, Basi di Dati. : McGraw Hill,

A. Albano, G. Ghelli, R. Orsini, Fondamenti di basi di dati. 2a Edizione.. : Zanichelli,

Eventuali indicazioni sui materiali di studio :

Verranno rese disponibili, come riferimento, le slide utilizzate a lezione.

CALCOLO NUMERICO

(Titolare: Prof.ssa MICHELA REDIVO ZAGLIA)

Periodo: Il anno, 2 semestre
Indirizzo formativo: Corsi comuni
Tipologie didattiche: 48A+16L; 7,00 CFU
Sede dell'insegnamento : Padova.

Prerequisiti :

Nozioni di base di Analisi Matematica, di Algebra lineare e Geometria (derivate di funzioni, spazi vettoriali, vettori, matrici, operazioni, determinanti, matrice inversa e matrici particolari, prodotto scalare, norme di vettori e di matrici).

Esami propedeutici: Algebra e Matematica Discreta, Analisi Matematica

Conoscenze e abilita' da acquisire :

Lo studente con questo corso

- avrÃ la possibilitÃ di acquisire capacitÃ informatiche numeriche di base;

- sarÃ in grado di costruire il modello numerico e l'algoritmo risolutivo di semplici problemi matematici;

- sarÃ in grado di programmare con il linguaggio di riferimento (Matlab) e produrre i risultati anche in forma grafica;

- acquisirÃ le conoscenze di alcuni metodi di base del Calcolo Numerico per applicazioni scientifiche e tecnologiche, con particolare attenzione ai concetti di errore, discretizzazione, approssimazione di dati o funzioni, convergenza, stabilitÃ , quadratura e integrazione di equazioni differenziali;

- sarÃ in grado di utilizzare i metodi presentati su esempi reali.

AttivitÃ di apprendimento previste e metodologie di insegnamento :

Il corso si articola in lezioni frontali ed esercitazioni in aula (48 ore) e lezioni in laboratorio informatico (16 ore) con esercitazioni sul computer in ambiente Matlab.

Molti dei metodi di base del Calcolo Numerico illustrati durante le lezioni, verranno via via utilizzati in laboratorio al fine di mostrare il loro effettivo utilizzo e le loro potenzialitÃ .

Numerosi esercizi di laboratorio verranno proposti agli studenti per integrare a casa il loro apprendimento.

Gradualmente lo studente prenderÃ dimestichezza con un ambiente di programmazione ed alla fine del corso dovrebbe essere in grado di superare un test che fa parte integrante dell'esame finale.

Durante il corso verranno effettuate attivitÃ di T4L (Teach for Learning) per migliorare l'apprendimento e l'interazione docente/studente e studente/studente.

Contenuti :

- Aritmetica del computer

I numeri: basi di numerazione e cambiamenti di base. Aritmetica del computer: rappresentazione dei numeri, operazioni macchina, errori, stabilitÃ e condizionamento.

- Equazioni non lineari

Metodi iterativi. Successioni convergenti. Metodo di bisezione. Metodi di punto fisso. Metodo di Newton. Test di arresto. Cenni sui sistemi non lineari.

- Algebra lineare numerica

Sistemi lineari; costo computazionale; errori e condizionamento; stime dell'errore; preconditionamento. Metodi diretti: Gauss, Cholesky, Householder (cenni). Fattorizzazioni LU e Cholesky. Calcolo del determinante e dell'inversa di una matrice. Metodi iterativi di rilassamento (Jacobi, Gauss-Seidel, SOR). Test di arresto.

Cenni al calcolo numerico degli Autovalori.

- Approssimazione discreta polinomiale

Interpolazione (Lagrange, Newton, Chebyshev). Minimi quadrati (retta di regressione).

- Quadratura numerica

Formule interpolatorie: Lagrange, Newton-Cotes, Gauss (cenni).

- Equazioni differenziali ordinarie

Metodi discreti ad un passo: Taylor, Eulero (implicito ed esplicito), Runge Kutta.

ModalitÃ di esame :

La prova d'esame Ã" divisa in due parti: prova scritta e test di laboratorio. Vi sono cinque appelli previsti: sessione estiva (due), sessione autunnale (due) e sessione invernale di recupero (uno).

- Prova scritta: Durante la prova scritta vengono proposti esercizi di Calcolo Numerico da svolgere a mano, con l'aiuto di una calcolatrice scientifica non programmabile, tenendo presente delle indicazioni fornite (uso di aritmetica esatta, uso di aritmetica approssimata, ...) e domande di comprensione a contenuto teorico. Ad ognuno degli esercizi e delle domande viene assegnato un punteggio il cui totale formerà il voto in trentesimi.

- Test di laboratorio: In ogni sessione, dopo la prova scritta, viene svolto un test nel laboratorio didattico informatico, i cui elaborati saranno corretti e valutati (insufficiente, sufficiente, buono, ottimo). Il test viene svolto in linguaggio Matlab e consiste nella risoluzione di un semplice problema di calcolo numerico con lo sviluppo di script, function ed eventuale produzione di grafici. Il superamento con valutazione sufficiente, buono o ottimo di tale test è prerequisite per superare l'esame.

- L'esame orale è possibile su richiesta, ma facoltativo.

Criteri di valutazione :

I criteri di valutazione con cui verrà effettuata la verifica delle conoscenze e delle abilità acquisite sono:

- 1 - Conoscenza dei vari metodi descritti sia dal punto di vista teorico ed algoritmico.
- 2 - Capacità di applicazione dei metodi appresi durante il corso a semplici esercizi di applicazione.
- 3 - Proprietà della terminologia matematica utilizzata e correttezza di risoluzione esatta dei calcoli.
- 4 - Abilità e dimestichezza nell'uso e nella scrittura di semplici programmi in ambiente Matlab.

Testi di riferimento :

Michela Redivo Zaglia, Calcolo Numerico: Esercizi. Padova: Libreria Progetto, 2015

Michela Redivo Zaglia, Calcolo Numerico: Metodi ed Algoritmi. Padova: Libreria Progetto, 2011

Eventuali indicazioni sui materiali di studio :

Tutto il materiale didattico integrativo del corso viene reso disponibile sulla piattaforma Moodle.

Esistono numerosi tutorial, manuali e corsi online gratuiti in rete relativi all'ambiente di programmazione Matlab.

Sul sito web del docente www.math.unipd.it/~michela

nella sezione didattica, è possibile recuperare alcuni link e le informazioni relative al recupero gratuito del software Matlab (l'Ateneo padovano ha acquisito una Licenza Campus)

INGEGNERIA DEL SOFTWARE

(Titolare: Prof. TULLIO VARDANEGA)

Periodo: III anno, annuale
Indirizzo formativo: Corsi comuni
Tipologie didattiche: 64A+8E+32L; 13,00 CFU

Prerequisiti :

L'insegnamento assume e richiede familiarità con i linguaggi e tecniche di programmazione presentati nel triennio (C, C++; programmazione imperativa e a oggetti), e con le funzionalità fondamentali delle base di dati e delle più comuni tecnologie basate su SQL.

L'importanza di tali conoscenze preliminari determina vincolo di propedeuticità per l'accesso alle attività didattiche di Ingegneria del Software, nel superamento degli insegnamenti Programmazione a oggetti e Basi di dati.

Conoscenze e abilità da acquisire :

L'insegnamento opera su due semestri consecutivi: nel primo eroga le conoscenze alla base della professione di "software engineer"; successivamente, nella prima metà del secondo mette gli studenti nella condizione di acquisire formazione pratica ed esperienziale, tramite lo svolgimento di un impegnativo progetto di gruppo attraverso tutte le fasi principali del suo ciclo di vita, dalla risposta a una gara di appalto fino alla revisione di accettazione e collaudo.

Nello specifico, lo studente apprenderà :

- 1) metodi e tecniche per l'organizzazione e la gestione di attività collaborative (pianificazione, ripartizione dei compiti, calendarizzazione, monitoraggio del progresso, verifica degli esiti);
- 2) professionalità nell'interazione con "clienti" esterni (cattura, analisi e discussione di requisiti, dimostrazione di prototipi e di prodotti finiti);
- 3) metodi e tecniche per l'auto-apprendimento e per la condivisione di conoscenza;
- 4) metodi e tecniche di presentazione e comunicazione.

Attività di apprendimento previste e metodologie di insegnamento :

- 1) insegnamento d'aula;
- 2) lezioni "rovesciate", svolte in modalità "flipped classroom", su argomenti selezionati, centrati prevalentemente su strumenti di sviluppo collaborativo;
- 3) esercitazione guidate su temi di teoria;
- 4) attività pratiche con feedback di progresso.

Contenuti :

Per una carrellata aggiornata degli argomenti di insegnamento, si veda la pagina <<http://www.math.unipd.it/~tullio/IS-1/2017/>>, che include diapositive di presentazione e materiale di approfondimento.

Modalità di esame :

L'esame di questo insegnamento consta di due parti complementari:

- una prova scritta individuale (calendarizzate a partire da metà aprile), il cui esito positivo conta per il 40% sul voto finale;
- una serie di revisione di avanzamento di progetto didattico, di gruppo, a partire dalla candidatura all'ingresso (a partire da metà gennaio), fino al collaudo del prodotto sviluppato (a partire da metà maggio), la cui media di esito conta per il rimanente 60% del voto finale individuale.

Criteri di valutazione :

Lo scritto individuale consiste di 3 domande pratiche di tipo "problem solving" e 3 domande sui concetti fondanti della disciplina del software engineering.

Nel complesso, le sei domande mirano a valutare il livello complessivo di apprendimento raggiunto dallo studente delle nozioni impartite durante il corso, sia in aula che nel progetto.

Per mettere alla prova le capacità di collaborazione sviluppate nel progetto didattico, due delle sei domande (una per ciascuno tipo) richiederanno risposta concordate da gruppi di due o tre studenti formati a caso, dal docente.

La valutazione delle revisioni di progresso, nel loro complesso, misura invece il gradiente di miglioramento qualità dei prodotti realizzati nel corso del progetto, sia documentali che software, e delle corrispondenti presentazioni e dimostrazioni pubbliche.

Testi di riferimento :

IEEE Computer Society. Software Engineering Coordinating Committee, Guide to the Software Engineering Body of Knowledge. : , 2014
Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, Design Patterns: Elements of Reusable Object-Oriented Software. :
Pearson Education, 1995

Ian Sommerville, Software Engineering (10th edition). : Pearson Education, 2015

Eventuali indicazioni sui materiali di studio :

Tutto il materiale didattico presentato durante le lezioni frontali Ã" reso disponibile su una apposita pagina-calendario dell'insegnamento, insieme a materiale di approfondimento.

LINGUA INGLESE

(Titolare: Prof. MASSIMO MARCHIORI)

Periodo: I anno, 2 semestre
Indirizzo formativo: Corsi comuni
Tipologie didattiche: ; 3,00 CFU
Sede dell'insegnamento : Vedi <http://corsi.math.unipd.it/inglese/>
Aule : Vedi <http://corsi.math.unipd.it/inglese/>

Contenuti :

Tutti i Corsi di Laurea di Scienze richiedono una conoscenza della Lingua inglese pari al livello B2 (abilitÃ ricettive ascolto e lettura) del Quadro Comune Europeo di Riferimento per le Lingue del Consiglio d'Europa.

Chi Ã" giÃ in possesso di una certificazione di livello B2 o superiore puÃ² chiederne il riconoscimento. Tutti gli altri studenti possono sostenere presso il Centro Linguistico di Ateneo il corrispondente Test di AbilitÃ Linguistica (TAL), il cui superamento permette il riconoscimento dei crediti formativi per la lingua straniera.

Tutte le informazioni sull'idoneitÃ , sul test di lingua e sulle certificazioni riconosciute, sono disponibili all'indirizzo http://www.scienze.unipd.it/index.php?id=inglese_triennali_1819

Per maggiori informazioni:
<http://corsi.math.unipd.it/inglese/>

Testi di riferimento :

CONTENUTO NON PRESENTE

LOGICA

(Titolare: Prof.ssa MARIA EMILIA MAIETTI)

Periodo: I anno, 1 semestre
Indirizzo formativo: Corsi comuni
Tipologie didattiche: 32A+18E; 6,00 CFU

Prerequisiti :

Nessuno

Conoscenze e abilita' da acquisire :

Lo scopo del corso Ã" fornire un'introduzione alla logica e alla sua rilevanza per l'informatica.

Lo studente dovrÃ quindi acquisire la capacitÃ di esprimere un enunciato tramite una formula di un linguaggio formale, di dare una dimostrazione tramite una derivazione in un sistema assiomatico e di fornire controesempi nel caso la formula non sia derivabile.

In particolare lo studente sarÃ condotto alla comprensione di alcuni concetti di carattere generale, come linguaggio, espressione, proposizione, asserzione, metalinguaggio, e altri specifici della matematica, come derivazione, dimostrazione, sistema assiomatico, interpretazione.

Lo studente raggiungerÃ padronanza di tali concetti e sarÃ quindi in grado di riconoscerli ed applicarli nella matematica, nell'informatica e anche nella vita quotidiana. Il corso illustrerÃ , inoltre, come la logica possa chiarire con rigore quali siano i limiti intrinseci a quel che puÃ² essere espresso in un dato linguaggio e a quel che puo' essere dimostrato in un dato sistema assiomatico. Infine, il corso darÃ qualche cenno alla storia della logica e alle sue potenzialitÃ e prospettive attuali.

AttivitÃ di apprendimento previste e metodologie di insegnamento :

Oltre alle lezioni teoriche sono previste esercitazioni in aula con risoluzione di molti esercizi. Sono previste simulazioni in aula degli esami scritti.

Contenuti :

1. Linguaggio, metalinguaggio, livelli di riferimento, iterazione infinita.

2. Concetto di macchina o robot, significato dei connettivi e loro regole di deduzione, calcolo dei sequenti per logica classica proposizionale, tabelle di verita', teoremi di validita' e completezza.

3. Calcolo dei sequenti per la logica classica predicativa, nozione di interpretazione e modello, teoremi di validita' e completezza.

4. Metodi di decisione per calcoli dei sequenti proposizionali classici.

5. Costruzione di contromodelli classici di enunciati predicativi.

6. Cenni ai teoremi di completezza e incompletezza (Gödel) e di indecidibilità (Church) e loro significato.

Modalità di esame :

Esame scritto

Criteri di valutazione :

La valutazione della prova scritta si baserà sull'assegnazione di un punteggio ad ogni esercizio presente nel testo d'esame.

Testi di riferimento :

Maria Emilia Maietti, Manuale pratico di Logica. : Padova, 2016

Giovanni Sambin, Per istruire un robot. : Libreria Cortina, Padova, 2007

Eventuali indicazioni sui materiali di studio :

Verranno fornite dispense contenenti tutti gli aspetti teorici e pratici necessari (con incluso l'assegnazione e lo svolgimento di esercizi specifici) per l'apprendimento degli argomenti trattati nelle lezioni frontali.

PROBABILITÀ E STATISTICA

(Titolare: Dott.ssa FRANCESCA COLLET)

Periodo: Il anno, 2 semestre
Indirizzo formativo: Corsi comuni
Tipologie didattiche: 32A+16E; 6,00 CFU

Prerequisiti :

Prerequisiti: Analisi matematica, Algebra e matematica discreta.

Conoscenze e abilità da acquisire :

Lo studente dovrà acquisire la conoscenza degli strumenti di base del calcolo delle probabilità e della statistica inferenziale. Alla fine del corso l'allievo dovrà essere in grado di costruire semplici modelli probabilistici di fenomeni aleatori e di effettuare i relativi calcoli.

Attività di apprendimento previste e metodologie di insegnamento :

Lezioni frontali. Nel corso delle lezioni sono esposti gli aspetti teorici del corso, vengono illustrati esempi di applicazione e svolte esercitazioni. Vengono inoltre proposte esercitazioni individuali da svolgere a casa con successiva illustrazione delle soluzioni.

Contenuti :

Calcolo delle probabilità. Assiomi e conseguenze elementari degli assiomi. Esempi di spazi di probabilità discreti; finiti; uniformi. Paradosso dei compleanni. Probabilità condizionata. Formula della probabilità totale e di Bayes. Eventi indipendenti. Variabili e vettori casuali discreti. Funzione di ripartizione e densità discreta. Variabili indipendenti. Parametri riassuntivi: valore atteso, varianza, momenti, correlazione, covarianza. Disuguaglianze: Jensen, Markov, Chebishev. Esempi di variabili casuali discrete: Bernoulli, binomiale, geometrica, Poisson. Teorema limite di Poisson. Variabili casuali assolutamente continue. Esempi: uniforme, esponenziale, Gaussiana. Legge dei grandi numeri (Chebishev). Il metodo di Montecarlo. Teorema limite centrale (Lindeberg-Lévy). Approssimazione normale.

Statistica Descrittiva. Dati qualitativi e quantitativi, frequenze relative, metodi grafici di analisi dei dati. Indici di centralità, di dispersione, e di forma. Correlazione tra caratteri numerici: retta di regressione, covarianza e coefficiente di correlazione

Statistica Inferenziale. Stimatori. Intervalli di confidenza. Test statistici: ipotesi e alternativa, regione critica, valore critico, errori di prima e seconda specie, potenza, il valore p, test bilateri e unilateri. Test sulla media e sulla differenza di medie. Test accoppiati. Stime e test per proporzioni: tabelle di contingenza e test chi quadro.

Modalità di esame :

Esame scritto (3 ore).

Criteri di valutazione :

Lo studente dovrà dimostrare di aver acquisito le conoscenze teoriche e di saperle applicare correttamente alla soluzione di problemi di calcolo delle probabilità e di statistica inferenziale di congrua difficoltà.

Testi di riferimento :

Lorenzo Finesso, Appunti di Probabilità. : ,

M. Bramanti, Calcolo delle probabilità e statistica. Bologna: Progetto Leonardo,

Eventuali indicazioni sui materiali di studio :

Tutti gli argomenti del corso vengono illustrati in aula. Gli appunti delle lezioni possono essere integrati dal libro di testo e dal materiale aggiuntivo reso disponibile sulla piattaforma moodle.

PROGRAMMAZIONE

(Titolare: Prof. GILBERTO FILE')

Periodo: I anno, annuale

Indirizzo formativo: Corsi comuni
Tipologie didattiche: 56A+24L; 10,00 CFU

Prerequisiti :

Qualche conoscenza di architettura degli elaboratori.

Conoscenze e abilità da acquisire :

Specificare un problema attraverso la formulazione di una pre- ed una post-condizione cui il programma che vogliamo costruire per risolvere il problema deve obbedire. La capacità di costruire un programma in C++ e dimostrare che fa quanto specificato nella sua pre- e post-condizione. Capacità di trovare anche soluzioni ricorsive ai problemi e di dimostrare la loro correttezza grazie ad una prova induttiva. Conoscenza delle nozioni di base della programmazione imperativa: comandi semplici, puntatori, array, funzioni, passaggio dei parametri per valore e riferimento, tipi definiti dall'utente, memoria dinamica, programmi su piA1 file, ed eccezioni.

Attività di apprendimento previste e metodologie di insegnamento :

Si segue l'approccio secondo cui per imparare a programmare in modo consapevole, A" necessario fare molti esercizi ricevendo feedback sulle soluzioni proposte. Quindi ogni settimana vengono assegnati esercizi che gli studenti sono invitati a risolvere. Gli esercizi sono resi disponibili attraverso un sito web basato su moodle sul quale gli studenti consegnano e testano le loro soluzioni rispetto ad alcuni input predisposti. In ogni momento ogni studente ha modo di verificare il suo progresso rispetto agli esercizi fatti. La consegna di soluzioni corrette (rispetto ai test) per alcuni degli esercizi assegnati, A" obbligatoria per accedere agli esami. Gli esami si svolgono nel laboratorio informatico e seguono le stesse modalità degli esercizi settimanali.

Il corso consiste di 80 ore totali di cui 24 in laboratorio informatico. Il corso verrà diviso in due parti: la prima parte di 2 crediti nel primo semestre, mentre il resto di 8 crediti si svolgerà nel secondo semestre. La prima parte ha lo scopo di rendere tutti gli studenti capaci di usare un PC per scrivere ed eseguire programmi e anche di presentare le prime nozioni di correttezza dei programmi.

Contenuti :

I contenuti del corso si pongono su due piani diversi:

1) Da una parte vengono insegnati alcuni concetti della correttezza dei programmi alla Hoare, cioA" basati sulle nozioni di pre-, post-condizione ed invarianti dei cicli. Ogni programma A" accompagnato da una pre- e post-condizione e la sua correttezza rispetto ad esse va dimostrata in modo convincente.

2) Contemporaneamente alla parte (1), vengono insegnate le nozioni di base della programmazione imperativa con C++. In particolare, tipi predefiniti, istruzioni semplici, puntatori, array, aritmetica dei puntatori, funzioni, funzioni ricorsive, memoria dinamica, liste concatenate ed alberi binari.

Modalità di esame :

L'esame consiste di una prova scritta con alcune domande teoriche ed un esercizio di programmazione. La parte di programmazione richiede di sviluppare un programma iterativo ed uno ricorsivo. Si richiede anche di specificare la correttezza delle funzioni prodotte.

L'esame si svolge in laboratorio informatico e gli studenti ricevono l'assegnamento e lavorano direttamente sul PC.

L'ammissione agli esami A" condizionata dalla consegna di alcuni esercizi assegnati durante il corso.

Criteri di valutazione :

L'esame A" fatto per mettere in rilievo la capacità di ragionare e di esprimere in forma chiara il proprio ragionamento, da parte dello studente. In particolare, si valuta la capacità di specificare il problema da risolvere e di spiegare i motivi per cui la soluzione proposta effettivamente risolve il problema considerato. Sono apprezzate semplicitA" e chiarezza.

Testi di riferimento :

Gilberto FilA", Programmazione consapevole. Padova: Progetto, 2014

Eventuali indicazioni sui materiali di studio :

Il corso usa un sito moodle che raccoglie tutto il materiale del corso: le regole del corso, le slide delle lezioni, gli esercizi settimanali che gli studenti possono risolvere e testare interattivamente, e finalmente l'accesso alle registrazioni di tutte le lezioni del corso.

PROGRAMMAZIONE AD OGGETTI

(Titolare: Prof. FRANCESCO RANZATO)

Periodo: Il anno, 1 semestre
Indirizzo formativo: Corsi comuni
Tipologie didattiche: 48A+20E+12L; 10,00 CFU
Sede dell'insegnamento : Dipartimento di Matematica Pura ed Applicata
Aule : Aula LuM250

Prerequisiti :

PropedeuticitA" : Programmazione.

Conoscenze e abilità da acquisire :

Il corso mira ad introdurre la programmazione orientata agli oggetti in tutti i suoi aspetti, incluso lo sviluppo di un progetto software.

Attività di apprendimento previste e metodologie di insegnamento :

L'insegnamento prevede lezioni frontali e lo sviluppo di un progetto software di laboratorio.

Contenuti :

Il corso introduce la programmazione orientata agli oggetti utilizzando il linguaggio C++. Si tratteranno i seguenti argomenti principali. Tipi di dato astratti. Classi e oggetti. Campi dati e metodi. Parti private e pubbliche. Costruttori. Overloading. Distruttori. Metodi e classi friend. Classi collezione. Tecniche di condivisione controllata della memoria. Template di funzioni e di classe. EreditarietA" e gerarchie di classi. Metodi virtuali. EreditarietA" multipla e derivazione virtuale. Classi e gestione delle eccezioni. Cenni di design pattern. Uso di alcune librerie standard e ausiliarie: libreria STL e classi contenitore, libreria di I/O, librerie grafiche (ad esempio, Qt). Da C++ a Java. Concetti di base della programmazione ad oggetti in Java. Il corso prevede un laboratorio in cui gli studenti realizzeranno un progetto di programmazione ad oggetti usando gli strumenti introdotti nel corso.

Modalità di esame :

Esame scritto: quesiti sulla modellazione di problemi mediante programmi ad oggetti; quesiti sul comportamento dei programmi ad oggetti.

Sviluppo di un progetto software orientato agli oggetti.

Eventuale esame orale di discussione del progetto software.

Criteri di valutazione :

L'esame scritto verte su tutti gli argomenti del corso. Il progetto di laboratorio sar  sviluppato in C++/Java ed utilizzer  alcune librerie ad ampia diffusione. L'esame orale consiste in una discussione del progetto.

Testi di riferimento :

Francesco Ranzato, Appunti di programmazione ad oggetti. Padova: Libreria Progetto, Padova, 2012

PROGRAMMAZIONE CONCORRENTE E DISTRIBUITA

(Titolare: Dott. ARMIR BUJARI)

Periodo: III anno, 1 semestre
Indirizzo formativo: Corsi comuni
Tipologie didattiche: 30A+18E; 6,00 CFU
Sede dell'insegnamento : Padova

Prerequisiti :

Conoscenze di programmazione.

E' propedeutico l'insegnamento di Programmazione ad Oggetti.

Conoscenze e abilita' da acquisire :

Il corso presenta il linguaggio Java e la programmazione a diversi livelli di astrazione: la programmazione ad oggetti, la programmazione funzionale, la programmazione concorrente, e la programmazione distribuita.

Attivita' di apprendimento previste e metodologie di insegnamento :

Il corso prevede lezioni teoriche, lezioni di esercitazioni, e la realizzazione di un progetto che prevede l'uso di tutte le caratteristiche del linguaggio Java viste nel corso.

Contenuti :

1. Breve ripasso alla programmazione ad oggetti in Java:

- classi, oggetti, ereditarieta', polimorfismo
- organizzazione delle classi: classi astratte, interfacce, classi interne (statiche, di istanza, anonime, innestate in interfacce)
- grafica e gestione degli eventi.

2. Programmazione funzionale in Java: Lambda e Stream API

3. Programmazione concorrente: thread, scheduling, accesso sincronizzato a dati condivisi, comunicazione tra thread.

4. Programmazione distribuita: stream e serializzazione, socket.

Modalita' di esame :

L'esame consiste in una prova scritta seguita opzionalmente dalla discussione orale dei contenuti del corso.

Criteri di valutazione :

La prova scritta valuta l'apprendimento del linguaggio java e la capacita' di realizzare soluzioni corrette per problemi di natura concorrente.

La prova orale valuta non solo la correttezza e la funzionalita' dell'applicazione distribuita realizzata, ma anche la capacita' dello studente di illustrare il programma e giustificarne le scelte di base.

Testi di riferimento :

Oracle, Java 8 Lambdas & Stream API. ; ,

Silvia Crafa, Oggetti, Concorrenza, Distribuzione. Programmare a diversi livelli di astrazione,.. : Editore Esculapio, 2014

PROVA FINALE

(Titolare: da definire)

Periodo: III anno, 2 semestre
Indirizzo formativo: Corsi comuni
Tipologie didattiche: ; 3,00 CFU
Sede dell'insegnamento : Informazioni in lingua non trovate
Aule : Informazioni in lingua non trovate

Prerequisiti :

CONTENUTO NON PRESENTE

Conoscenze e abilita' da acquisire :

CONTENUTO NON PRESENTE

Attivita' di apprendimento previste e metodologie di insegnamento :

CONTENUTO NON PRESENTE

Contenuti :

CONTENUTO NON PRESENTE

Modalita' di esame :

CONTENUTO NON PRESENTE

Criteri di valutazione :

CONTENUTO NON PRESENTE

Testi di riferimento :

CONTENUTO NON PRESENTE

Eventuali indicazioni sui materiali di studio :

CONTENUTO NON PRESENTE

RETI E SICUREZZA

(Titolare: Prof. MASSIMO MARCHIORI)

Periodo: Il anno, 1 semestre
Indirizzo formativo: Corsi comuni
Tipologie didattiche: 72A+8L; 10,00 CFU

RICERCA OPERATIVA

(Titolare: Dott. LUIGI DE GIOVANNI)

Periodo: III anno, 1 semestre
Indirizzo formativo: Corsi comuni
Tipologie didattiche: 36A+12E+12L; 7,00 CFU
Sede dell'insegnamento : Padova

Prerequisiti :

Conoscenze di base di analisi matematica e algebra.
E' propedeutico l'insegnamento di "Algebra e Matematica Discreta".

Conoscenze e abilità da acquisire :

Costruzione di modelli matematici per il supporto alle decisioni e relativi algoritmi, con particolare riferimento alla programmazione lineare nel continuo e nel discreto e all'ottimizzazione su grafi. Uso di pacchetti software per la soluzione di problemi di ottimizzazione.

Attività di apprendimento previste e metodologie di insegnamento :

L'insegnamento prevede lezioni frontali ed esercitazioni in laboratorio. Le esercitazioni in laboratorio consistono nell'implementazione in un linguaggio di modellazione algebrica di semplici modelli di programmazione lineare (mista intera).

Contenuti :

1. Problemi di ottimizzazione e modelli: modellazione e utilizzo di risolutori software in laboratorio.
2. Programmazione lineare: teoria e metodo del simplesso, teoria della dualità e applicazioni.
3. Ottimizzazione su grafi: modelli e algoritmi per il problema dell'albero di copertura di costo minimo, il problema del cammino minimo (algoritmi di Dijkstra e Bellman-Ford), il problema del flusso massimo (algoritmo di Ford-Fulkerson) e del flusso di costo minimo.
4. Elementi di Programmazione Lineare Intera e Ottimizzazione Combinatoria: metodi esatti (Branch-and-Bound), cenni su metodi euristici e metaeuristici (ricerca locale e varianti).

Modalità di esame :

L'esame scritto, comprensivo di un problema da formulare con un modello di programmazione lineare, esercizi e domande di teoria. Se necessario, si terrà una discussione orale dello scritto. Il candidato può inoltre, a sua discrezione, preparare un mini-progetto facoltativo.

Criteri di valutazione :

L'esame scritto richiede lo svolgimento di esercizi per la valutazione del livello di apprendimento degli argomenti svolti (ad esempio, modellazione di un problema di ottimizzazione in programmazione lineare intera, applicazione dell'algoritmo del simplesso, applicazione di algoritmi di ottimizzazioni su rete, applicazione della teoria della dualità, applicazione dell'algoritmo del Branch-and-Bound, domande sui diversi argomenti svolti etc.)

Testi di riferimento :

CONTENUTO NON PRESENTE

Eventuali indicazioni sui materiali di studio :

Vengono rese disponibili delle dispense degli argomenti trattati a lezione e dei lucidi degli argomenti trattati in laboratorio, che contengono tutte le nozioni richieste all'esame.

Gli studenti interessati possono approfondire gli argomenti sui seguenti testi:

- M. Fischetti, *Lezioni di Ricerca Operativa*, 1999, Libreria Progetto Padova.
- D. Bertsimas, J. Tsitsiklis, *Introduction to linear optimization*, 1996, Athena Scientific.
- R. K. Ahuja, T. L. Magnanti, J. B. Orlin "Network flows. Theory, algorithms, and applications", 1993, Prentice Hall.
- L. A. Wolsey: "Integer programming", 1998, Wiley.

SISTEMI OPERATIVI

(Titolare: Prof. CLAUDIO ENRICO PALAZZI)

Periodo: I anno, 2 semestre
Indirizzo formativo: Corsi comuni
Tipologie didattiche: 40A+24E+8L; 9,00 CFU

Prerequisiti :

Gli studenti dovrebbero preferibilmente avere una conoscenza generale delle Architetture dei Computer, così come fornita nel corso di Architettura degli Elaboratori.

Tuttavia, l'insegnamento non prevede propedeuticità.

Conoscenze e abilità da acquisire :

Questo corso introduce alle funzionalità di base dei moderni sistemi operativi. In particolare, il corso è diviso in tre parti principali. Nella prima, allo studente vengono presentati argomenti quali processi e thread, scambi di contesto, sincronizzazione, ordinamento e stallo. Nella seconda parte del corso, lo studente impara a conoscere problematiche e possibili soluzioni riguardanti la gestione della memoria quali, ad esempio, allocazione dinamica della memoria, memoria virtuale, paginazione e segmentazione. La terza parte del corso tratta il file system, inclusa la gestione di dischi e partizioni. Il corso termina con un'analisi delle scelte progettuali effettuate da sistemi operativi esistenti in commercio.

Attività di apprendimento previste e metodologie di insegnamento :

L'insegnamento prevede lezioni frontali, esercitazioni in aula e in laboratorio.

Contenuti :

Introduzione ai Sistemi Operativi.

Gestione dei Processi: definizione, strutture, concorrenza, sincronizzazione, ordinamento, stallo.

Gestione della Memoria: gerarchie, rilocalizzazione, strutture, memoria virtuale, paginazione, segmentazione.

File System: architetture, struttura logica, modalità di accesso, directory, aspetti implementativi.

Modelli e Architetture di Sistemi Operativi: discussione sulle scelte progettuali dei sistemi UNIX/Linux e dei sistemi Windows.

Modalità di esame :

Lo studente deve superare un esame scritto.

Criteri di valutazione :

Lo scritto contiene domande ed esercizi che consentono di valutare il livello di apprendimento delle nozioni discusse in classe e l'abilità dello studente nel maneggiare concetti in modo pratico.

Testi di riferimento :

A. S. Tanenbaum, *Modern Operating Systems - 4th Edition.* : Prentice Hall, 2014

Eventuali indicazioni sui materiali di studio :

Vengono rese disponibili le trasparenze utilizzate a lezione

STAGE

(Titolare: Prof. TULLIO VARDANEGA)

Periodo: III anno, 2 semestre

Indirizzo formativo: Corsi comuni

Tipologie didattiche: ; 11,00 CFU

TECNOLOGIE OPEN-SOURCE

(Titolare: Dott. NICOLA BERTAZZO)

Periodo: III anno, 1 semestre

Indirizzo formativo: Corsi comuni

Tipologie didattiche: 48A; 6,00 CFU

Prerequisiti :

Nessuno.

Conoscenze e abilità da acquisire :

Fornire agli studenti un bagaglio di esperienza base per la gestione tecnologica di un progetto software e la definizione e l'implementazione di una continuous delivery pipeline.

Attività di apprendimento previste e metodologie di insegnamento :

Lezioni frontali e laboratorio.

Contenuti :

Saranno trattati i seguenti temi:

- Issue Tracking
- Source Code Management (SCM)
- Testing Software
- Processo di Build
- Continuous integration e Continuous Delivery
- Configuration Management

Modalità di esame :

Orale.

Criteri di valutazione :

La valutazione si baserà su colloqui (individuali o di gruppi), a valle di 4 assignment presentati e discussi ai laboratori.

Testi di riferimento :

CONTENUTO NON PRESENTE

Eventuali indicazioni sui materiali di studio :

Slide e materiale indicato nelle slide quando necessario.

TECNOLOGIE WEB

(Titolare: Dott. LAMBERTO BALLAN)

Periodo: III anno, 1 semestre

Indirizzo formativo: Corsi comuni

Tipologie didattiche: 40A+12E+20L; 9,00 CFU

Prerequisiti :

È opportuno avere familiarità con gli elementi di base della programmazione, così come forniti nei corso di "Programmazione" e "Programmazione ad oggetti".

Gli studenti devono aver superato il corso di "Basi di Dati"

Conoscenze e abilità da acquisire :

L'insegnamento intende presentare agli studenti il World-Wide Web e le tecnologie informatiche che lo caratterizzano. Ha lo scopo di fornire le conoscenze necessarie per la progettazione e lo sviluppo di siti web di qualità con l'uso delle tecnologie più avanzate. Gli studenti, oltre ad acquisire una conoscenza di alto livello dei vari tipi di tecnologie web esistenti, verranno formati a divenire sviluppatori di siti web basati sui linguaggi standard e la tecnologia XML. Verranno inoltre trattati aspetti dell'interattività sul web (linguaggi di script).

Attività di apprendimento previste e metodologie di insegnamento :

L'insegnamento prevede lezioni frontali, esercitazioni in laboratorio e la realizzazione di un progetto.

Contenuti :

1. Introduzione. Il concetto di ipertesto, il World Wide Web ed Internet. Gli enti di standardizzazione, le architetture Client-Server e i protocolli di Internet.
2. I linguaggi del web statico. I linguaggi XHTML e HTML5 e i fogli stile (il linguaggio CSS): formattazione del testo e la grafica su Web; links e navigazione.
3. Principi di web design. Architettura dell'informazione. Schemi Organizzativi e strutture per la navigazione. Progettazione dell'interfaccia. Accessibilità e legislazione. Tecniche per garantire l'accessibilità. Search Engine Optimization.
4. Il linguaggio XML. Extensible Markup Language (XML), i linguaggi per la definizione di uno schema (DTD e XMLSchema), cenni al reperimento dati (XPath) e introduzione ai fogli di trasformazione di stile per XML (XSLT).
5. I linguaggi per il web dinamico (Programmazione su Internet) sia lato client che lato server. Il linguaggio Javascript. Il modello DOM per la gestione delle pagine via JavaScript. Il linguaggio PHP.

Modalità di esame :

Lezioni frontali, esercitazioni in laboratorio, e realizzazione di un progetto.

Criteri di valutazione :

Lo scritto contiene alcune domande che consentono di valutare il livello di apprendimento delle nozioni teoriche impartite durante il corso.

Il progetto, svolto in gruppo, mira a valutare la capacità, da parte dello studente, di individuare un caso di studio adeguato, e di progettare e realizzare un sito web sia per quanto riguarda la parte di backend che di frontend.

Testi di riferimento :

CONTENUTO NON PRESENTE

Eventuali indicazioni sui materiali di studio :

I lucidi del corso e il materiale dei laboratori sono messi a disposizione sul sito web del corso.