

Corso di Laurea in Informatica

Il mondo è sempre più informatizzato. Non esiste oramai attività commerciale, industriale, tecnica, scientifica che possa affermarsi, crescere, raggiungere risultati senza l'informatica.

E siamo solo all'inizio, probabilmente, di un lunghissimo periodo di espansione delle applicazioni di questa scienza, che studia le modalità di trattamento dei dati, la loro trasformazione in informazione e quindi in conoscenza. Già ora, comunque, tra le più grandi aziende del mondo compaiono quelle che producono software, oppure strumenti e servizi per il web. O ancora, come nel caso delle case farmaceutiche o delle aziende di telecomunicazioni, altre imprese non potrebbero esistere senza un continuo aggiornamento degli strumenti informatici, sia hardware che software. In effetti, l'intera economia attuale, così come larga parte delle attività umane svolte nell'ambito delle società avanzate, sono strettamente dipendenti dall'utilizzo delle tecnologie informatiche: non a caso, è ormai invalso l'uso del termine "società dell'informazione" per designare il contesto sociale ed economico in cui viviamo. In tale contesto, lo sviluppo delle nostre società, in tutte le loro componenti, è condizionato dalla capacità di fare uso di tecnologie informatiche, e la disponibilità di esperti informatici, in grado di operare ed apportare innovazione nello sviluppo e nell'utilizzo di tali tecnologie appare un elemento destinato ad assumere un ruolo cruciale ai fini di tale sviluppo. Studiare informatica – e diventare un bravo informatico – non significa però semplicemente imparare a installare del software e a utilizzare un computer. Come diceva Edsger Dijkstra, un grande "computer scientist" olandese, tra informatica e computer c'è la stessa relazione che tra astronomia e telescopio: mentre l'informatica è una scienza, che poggia i suoi principi nella matematica, il computer è solo uno strumento che consente di osservare tali principi e il risultato della loro applicazione. Ma essere un informatico non significa nemmeno "smanettare" con la tastiera nel chiuso di una stanza con la luce dello schermo eternamente proiettata sulla faccia. È qualcosa di più importante, utile e ricco di rapporti con altre persone, con altri esperti.

È soprattutto sapere partire da un problema e, analizzandolo, inventare il modo più efficiente di risolverlo; è poi anche, a partire da questo, realizzare delle applicazioni informatiche che consentano la risoluzione "automatica" del problema stesso mediante un computer. È trovare il modo per far sì che decine o centinaia, se non migliaia, di esperti possano collaborare, ognuno per la sua parte, nella realizzazione di un'unica applicazione software, e che questa applicazione possa, alla fine, fare esattamente ciò che ci attendeva inizialmente.

Per tutto ciò è necessario conoscere non soltanto la programmazione ma, soprattutto, avere acquisito le capacità di "risolutori di problemi", un mix di creatività e di rigore scientifico, che consente di individuare, formalizzare matematicamente e valutare modalità di soluzione di problemi. È poi necessario conoscere i fondamenti teorici che sono alla base della progettazione del software, in modo da poterli utilizzare al meglio e da essere capace di adattarli a esigenze diverse. Informatica "fondamentale" e informatica "pratica" sono entrambe indispensabili per poter essere un protagonista della grande rivoluzione tecnologica del nostro tempo.

Se si conoscono bene le basi di questa scienza, si è capaci di affrontare un cambiamento che in questo settore, come tutti sanno, è rapidissimo: nel 1964, il fisico Gordon Moore affermò che "la velocità dei processori raddoppia ogni 18 mesi". Oltre quarant'anni dopo, è ancora vero. Le macchine cambiano, evolvono, e con esse cambia la difficoltà dei problemi che mediante esse si intende risolvere e la complessità delle applicazioni e dei sistemi informatici che vengono realizzati per affrontare tali problemi: se si vuole essere protagonisti bisogna avere delle conoscenze forti. E i laureati in informatica sono protagonisti senza ombra di dubbio. Chi conosce i fondamenti di questa disciplina è molto richiesto, ad esempio, nei luoghi della ricerca scientifica e dell'innovazione in tutti i settori: perché oggi, ad esempio, la genetica non potrebbe progredire senza utilizzare continuamente nuove soluzioni informatiche. E lo stesso vale per la fisica delle particelle, per l'astrofisica, per le nanotecnologie, eccetera. Ma i laureati in informatica sono indispensabili anche là dove si decidono i destini futuri della produzione di massa dai vestiti alle automobili, dai cellulari agli oggetti dello sport, come il pallone o la racchetta – e di quella di altissima qualità (i farmaci d'avanguardia, i nuovi prodotti chimici, i sistemi di telecomunicazioni, le barche da regata, le automobili o le moto da competizione). Un'altra tipologia di lavoro è quella legata ai servizi: il commercio elettronico, la sicurezza informatica, l'innovazione informatica e

telematica dell'amministrazione pubblica. Si pensi solo a che cosa significa la diffusione a tutti gli italiani della carta d'identità elettronica. L'informatica è importantissima poi anche per la capacità di sviluppare la rete, il web, farlo evolvere, trasformarlo sempre più in uno strumento per lo scambio di informazioni e di idee, indispensabile per la creazione di una vera società globale, e qui si pongono tutte le attività inerenti, tra l'altro, la produzione di strumenti e servizi informativi per internet: dai siti dei mass media (giornali, televisioni, radio...) a quelli aziendali, culturali, sportivi, alle più recenti e sofisticate applicazioni e servizi di collaborazione mediata da internet, di social networking, al web semantico, con le sue relazioni con l'intelligenza artificiale. Il corso di laurea a Tor Vergata tiene conto di tutto questo e punta a formare degli specialisti informatici che abbiano acquisito i fondamenti della cultura informatica e che, essendo in grado di applicare tali principi, si pongano come specialisti ad uno stesso tempo competenti e flessibili in settori quali la progettazione, lo sviluppo e la gestione delle applicazioni software. Il percorso di studi prevede, quindi, sia un insegnamento teorico che uno più pratico.

Questo significa ovviamente molto spazio al laboratorio, alle esercitazioni con e senza tutor e, per preparare all'ingresso nel mondo del lavoro, gli stage. Al tempo stesso, viene prestata attenzione a favorire l'acquisizione di una adeguata familiarità con il metodo scientifico e di indagine, oltre che allo sviluppo di una certa sensibilità rispetto agli aspetti organizzativi ed economici insiti nella produzione di prodotti e servizi. A Tor Vergata è previsto che una parte dei corsi sia determinata a scelta dello studente.

Il progetto finale, necessario per conseguire la laurea, può essere svolto, su richiesta dello studente, in aziende, laboratori e strutture sia in Italia che all'estero. Alla fine dei tre anni, si ha la capacità di usare la tecnologia informatica per affrontare e risolvere problemi, così come per realizzare nuovi strumenti e servizi informatici. Dopo la laurea, lo studente può scegliere se approfondire la sua formazione, continuando gli studi per la laurea specialistica, oppure entrare subito nel mondo del lavoro. Il corso di laurea in informatica è a numero programmato: questo è necessario per poter garantire agli studenti una qualità elevata della formazione. Per questo, occorre superare una prova di ammissione per iscriversi.

Secondo le nuove disposizioni (D.M. 270/04), al fine di ottenere la Laurea in Informatica è necessario acquisire 180 crediti formativi (CFU) nell'ambito delle varie attività didattiche. L'attività formativa prevede insegnamenti teorici e pratici suddivisi in moduli didattici di base, moduli didattici caratterizzanti, moduli didattici di materie affini o integrative, moduli didattici concernenti attività formative complementari e stage volti a favorire l'inserimento lavorativo dello studente.

Tutti i percorsi formativi danno ampio spazio a esercitazioni e ad attività di tutorato e di laboratorio.

I crediti relativi alle attività didattiche di base, caratterizzanti, e affini o integrative sono acquisiti seguendo moduli didattici, e superando i relativi esami, secondo il piano delle attività formative ed in base alla programmazione didattica definiti dal Consiglio di Corso di Laurea (CCL) e pubblicati sul sito web di Informatica <http://www.informatica.uniroma2.it/index.htm>.

I crediti relativi alle attività a scelta dello studente vengono normalmente acquisiti da parte dello studente mediante la frequenza di insegnamenti scelti da un elenco predisposto dal CCL ed il superamento dei relativi esami. Lo studente effettua tale scelta presentando al CCL un piano di studio, secondo le modalità previste. L'acquisizione di tali crediti mediante differenti meccanismi verrà sempre valutata dal CCL in riferimento agli obiettivi formativi del corso di laurea ed alla valenza culturale complessiva del piano di studio proposto ed eventualmente alla carriera pregressa dello studente. La lingua straniera considerata è, salvo motivate eccezioni, l'inglese. I crediti relativi alle altre attività formative di cui all'Articolo 10, comma 1, lettera F del DM 509/99 ("attività formative per acquisire ulteriori conoscenze linguistiche, nonché abilità informatiche e telematiche, relazionali, o comunque utili per l'inserimento nel mondo del lavoro") sono attribuiti nell'ambito di opzioni individuate dal CCL.

Tali opzioni possono prevedere: frequenza di corsi brevi o cicli di seminari su tematiche specifiche: il CCL darà preventiva informazione di tali iniziative, specificando i crediti relativi e le modalità per la loro acquisizione; effettuazione di attività supplementari, assegnate su base individuale, nell'ambito di insegnamenti o della prova finale. Sul sito web di Informatica <http://www.informatica.uniroma2.it/index.htm> si potrà trovare l'elenco degli insegnamenti attivati (con i relativi crediti e con l'indicazione dell'attività formativa di riferimento), che permettono allo studente di realizzare gli obblighi formativi prescritti, oltre alle eventuali propedeuticità tra i vari

insegnamenti.

Modalità o requisiti di ammissione al Corso di Laurea

Gli aspiranti debbono, nei tempi prestabiliti presentare domanda di iscrizione al Corso di Laurea, secondo le modalità definite dall'Ateneo e disponibili sul sito web di ateneo. Al fine di poter erogare attività formative in maniera adeguata, il Corso di Laurea in Informatica prevede un numero programmato delle iscrizioni. Informazioni sulle modalità di svolgimento della prova di ammissione sono disponibili sul sito web di ateneo.

Prova finale

Per essere ammessi alla prova finale bisogna avere acquisito 174 crediti maturati mediante il superamento delle prove didattiche previste dal proprio piano di studi. La prova finale per il conseguimento della Laurea in Informatica richiede da parte dello studente l'approfondimento di un argomento affine al contenuto di almeno un corso presente nel proprio piano formativo. Tale argomento deve essere concordato con il docente del corso, che svolge le funzioni di relatore. L'argomento trattato deve essere svolto come progetto documentato mediante elaborato scritto. Modalità diverse di prova finale possono essere autorizzate dal CCL, sulla base di una richiesta motivata. In particolare, in relazione ad obiettivi specifici, e nel quadro di convenzioni che lo prevedono esplicitamente lo svolgimento della tesi può essere effettuato mediante tirocini formativi presso aziende, strutture della pubblica amministrazione ed enti esterni, oltre che nell'ambito di soggiorni di studio presso altre Università italiane ed estere, anche nel quadro di accordi internazionali. In ogni caso lo studente deve redigere un documento scritto e sostenere una prova seminariale. In questa prova il candidato dovrà mostrare, oltre alla padronanza dell'argomento trattato, autonomia e capacità espositiva e di ricerca bibliografica. Le prove finali si svolgono di norma in quattro appelli annuali, specificati ogni anno sul sito web di Informatica <http://www.informatica.uniroma2.it/index.htm>. La Commissione esprime un voto in centodecimi: agli studenti che raggiungono il punteggio di 110 può essere attribuita la lode.

Piano di studio

Durante il secondo anno di corso, lo studente deve presentare un piano di studio che dovrà essere approvato dal Consiglio di Corso di Laurea. A tal fine il Consiglio di Corso di Laurea pubblica ogni anno, sito web di Informatica <http://www.informatica.uniroma2.it/index.htm>, degli schemi di piani di studio consigliati. Piani di studio che rispettino le condizioni specificate in tali schemi sono automaticamente accettati dal Consiglio di Corso di Laurea.

Piano delle attività formative

Nel seguito viene presentato il piano delle attività formative, secondo le nuove norme.

Ordinamento degli Studi - Laurea Triennale

1 ANNO

Programmazione dei calcolatori con laboratorio	12 crediti	1° e 2° Semestre
Reti logiche ed architettura dei calcolatori	12 crediti	1° e 2° Semestre
Analisi Matematica	9 crediti	1° Semestre
Matematica discreta	6 crediti	1° Semestre
Elementi di matematica	3 crediti	1° Semestre
Fondamenti di informatica	6 crediti	2° Semestre
Fisica	6 crediti	2° Semestre
Geometria ed algebra	6 crediti	2° Semestre

2 ANNO

Algoritmi e strutture di dati con laboratorio	6 crediti	1° Semestre
Basi di dati	6 crediti	1° Semestre
Linguaggi di programmazione con laboratorio	6 crediti	1° Semestre

Statistica per la ricerca sperimentale e tecnologica	6 crediti	1° Semestre
Ricerca operativa	6 crediti	1° Semestre
Architettura dei Calcolatori	6 crediti	2° Semestre
Laboratorio di Basi di dati	6 crediti	2° Semestre
Metodologie di programmazione con laboratorio	6 crediti	2° Semestre
Reti di calcolatori	6 crediti	2° Semestre
Sistemi operativi con laboratorio	6 crediti	2° Semestre
Fisica 1	3 crediti	2° Semestre
3 ANNO		
Algoritmi e strutture di dati 2	6 crediti	1° Semestre
Calcolo Numerico	6 crediti	1° Semestre
Inglese	3 crediti	1° Semestre
Ingegneria del Software	6 crediti	1° Semestre
Intelligenza artificiale	6 crediti	1° Semestre
Linguaggi e Traduttori	6 crediti	2° Semestre
Fisica 2	3 crediti	2° Semestre
Laboratorio Sistemi software	6 crediti	2° Semestre
Architettura dei Sistemi Software	6 crediti	2° Semestre
Analisi matematica 2	6 crediti	2° Semestre
Architetture software	5 crediti	2° Semestre
per applicazioni in tempo reale ad alta interattività		

Programmi dei Corsi

Programmazione dei calcolatori con laboratorio

cfu 12 semestre 1/2

Dr. Bianchi

Introduzione alla programmazione: Il modello di calcolo. Storia e caratterizzazione dei linguaggi di programmazione. Nozioni teoriche di teoria della programmazione introdotte utilizzando le istruzioni di base di un linguaggio interpretato. Diagrammi di flusso. Programmazione strutturata in C: espressioni, istruzioni per il controllo del flusso, array, stringhe, puntatori, funzioni, passaggio dei parametri, array, puntatori, allocazione dinamica della memoria, gestione dell'I/O. Algoritmi e applicazioni: principali algoritmi di ordinamento, introduzione e utilizzo delle strutture dati di base e implementazione code, stack, liste, alberi. Uno sguardo ad altri paradimi di programmazione: il paradigma funzionale, orientato agli oggetti e logico.

Reti logiche ed architettura dei calcolatori

cfu 12 semestre 1/2

Dr. Gualà

Il corso propone un percorso che parte dalla rappresentazione dell'informazione, passa attraverso la logica dei sistemi digitali e arriva a definire i fondamentali concetti architettureali, mostrandone la pratica traduzione in termini di logica e di organizzazione della macchina. Più in dettaglio, l'insegnamento è diviso in due parti. La prima parte approfondisce le tematiche relative alla progettazione di reti logiche combinatorie e sequenziali, mentre la seconda verte sulla struttura interna dei calcolatori (CPU, Memoria, dischi e altri dispositivi). Viene inoltre presentato il linguaggio Assembly.

Analisi Matematica

cfu 9

semestre 1

Prof. Radulescu

Richiami di insiemistica elementare, trigonometria e geometria elementare. Numeri naturali ed interi, numeri razionali. Numeri irrazionali, calcolo di radice di due. Concetto di limite. Successioni di Cauchy. Numeri reali. Calcolo di limiti, limiti notevoli. Il numero "e". Logaritmi, definizione e proprietà. Principio di induzione. Serie numeriche. Condizione necessaria per la convergenza. Criteri di convergenza per serie a termini alterni (Leibniz) e serie a termini positivi (confronto, rapporto, radici). Convergenza assoluta. funzioni, operazioni su funzioni, concetto di grafico, limiti di funzioni. continuità. criterio per l'esistenza di zeri di funzioni continue in un intervallo chiuso e limitato. Metodo di bisezione. Sup e inf di insiemi numerici. esistenza massimi di funzioni

continue, estrazione di una sottosuccessione convergente da una successione contenuta in un intervallo chiuso e limitato. derivate (interpretazione geometrica e cinematica). Regole per il calcolo delle derivate. Derivate notevoli. Teoremi di Rolle e Lagrange. grafici di funzioni. Formula di Taylor con resto di Lagrange. Algoritmo di Newton.

Matematica discreta

cfu 6 semestre 1

Prof. Piacentini

Relazioni d'ordine e di equivalenza. Insiemi quoziente. Cardinalita' di insiemi. Combinatoria. Il principio di inclusione-esclusione. Proprieta' degli interi. Relazioni ricorsive. Congruenze modulo n. Sistemi di congruenze e il teorema cinese dei resti. Crittografia a chiave pubblica: il sistema RSA. Polinomi in una indeterminata. Questioni di irriducibilita' su vari campi. Campi finiti e loro costruzione. Le fondamentali strutture algebriche: gruppoidi, semigruppoidi, monoidi, gruppi, anelli, campi. Il gruppo simmetrico

Elementi di matematica

cfu 3 semestre 1

Prof. Piacentini

I numeri naturali, i numeri interi, i numeri razionali. Espressioni numeriche e letterali. Prodotti notevoli. Fattorizzazione. Disuguaglianze e loro proprieta'. Insiemi e operazioni tra insiemi. Polinomi in una o piu' indeterminate. Disequazioni di primo e secondo grado. Equazioni con valore assoluto e loro risoluzione.

Elementi di logica. Funzioni. Induzione matematica. Radicali e loro proprieta'. Logaritmi.

Fondamenti di informatica

cfu 6 semestre 2

Prof. Giammarresi

Introduzione alla teoria della computazione. Linguaggi regolari. Definizione di linguaggio formale e notazioni.

Automa a stati finiti deterministico. Automa a stati finiti non deterministico. Teorema di equivalenza di automa non deterministico con automa deterministico. Espressioni regolari. Pumping Lemma. Proprieta' di chiusura. Algoritmi per automi a stati finiti. Minimizzazione di automi. Il problema di string matching e i relativi algoritmi basati su automi a stati finiti. Strumenti di manipolazione dei testi basati su automi a stati finiti ed espressioni regolari: grep e programmi derivati (awk, sed, perl). Linguaggi context-free. Grammatiche context-free. Derivazioni. Automi a pila. Teorema di equivalenza grammatiche context-free / automi a pila. Algoritmo di riconoscimento di Cocke-Kasami-Young. Proprieta' di chiusura. Linguaggi non context-free. Forma Normale di Chomsky. Determinismo. Definizione macchina di Turing. Computazioni di macchine di Turing. Estensioni di macchine di Turing. Linguaggi ricorsivamente enumerabili e linguaggi ricorsivi. La tesi di Turing-Church. Macchine di Turing Universali. Il problema della fermata. Problemi indecidibili.

Fisica

cfu 6 semestre 2

Prof. Goletti

Vettori e scalari. Prodotto scalare. Prodotto vettoriale. Moto rettilineo. Moto in due e tre dimensioni. Le equazioni del moto uniformemente accelerato. Equazione della traiettoria. Moto circolare uniforme. Accelerazione centripeta. Le forze ed il moto. Il principio di inerzia. La forza. Seconda legge di Newton. Terza legge di Newton. Energia cinetica e lavoro. Lavoro svolto da una generica forza variabile. Teorema dell'energia cinetica (o delle forze vive). Energia potenziale e conservazione dell'energia meccanica. Forze conservative. Bilancio dell'energia meccanica in presenza di forze non conservative. Lavoro esterno su un sistema svolto da una forza. Sistemi di punti materiali. Il centro di massa. Moto del centro di massa. Equazione di Newton per un sistema di punti materiali. Quantità di moto. Quantità di moto di un sistema di punti materiali. Conservazione della quantità di moto. Urti. La carica elettrica. Il campo elettrico. Principio di sovrapposizione. Flusso del campo elettrico. Legge di Gauss. Potenziale elettrico. Campi conservativi. Energia potenziale elettrica di un sistema di cariche. Come calcolare il campo elettrico noto il potenziale. Capacità elettrica. Il condensatore. Condensatore in presenza di un dielettrico. Legge di Gauss per un dielettrico.

Geometria ed algebra

cfu 6 semestre 2

Prof. Letizia

Sistemi di equazioni lineari. Metodo di eliminazione di Gauss. Matrici. Somma e prodotto di matrici. Matrici invertibili: Determinante: Spazi vettoriali. Sottospazi: Somme e Intersezioni di sottospazi: Dipendenza e indipendenza lineare: Generatori basi dimensione. Formula di Grassmann: Rango di matrici. Coordinate. Cambiamenti di base e di coordinate.

Trasformazioni lineari: Nucleo e Immagine di una trasformazione lineare: Composizione di trasformazioni lineari: Matrice associata a trasformazione lineare, suo cambiamento al variare delle basi. Endomorfismi autovalori autovettori polinomio caratteristico: Diagonalizzabilità. Spazio duale. Forme bilineari simmetriche e hermitiane: Teorema di inerzia: Spazi con prodotto scalare. Procedimento di Gram-Schmit. Operatore aggiunto.

Operatori e matrici ortogonali unitarie e simmetriche:

proprietà dei loro autovalori ed autospazi: Esistenza di basi ortonormali di autovettori per operatori unitari e simmetrici

2 ANNO

Algoritmi e strutture di dati con laboratorio

cfu 6

semestre 1

Dr. Rossi

Proseguimento del corso di "Elementi di Algoritmi e Strutture Dati" del primo anno. Vengono affrontati problemi su grafi e viene introdotta la teoria della complessità computazionale. In particolare gli argomenti trattati sono: i grafi e le loro proprietà; algoritmi di visita di grafi; algoritmi per il calcolo delle componenti connesse, componenti fortemente connesse e cicli; problemi di ottimizzazione su grafi tra cui il problema del minimo albero ricoprente e dei cammini minimi; introduzione alla teoria della complessità, i problemi NP-completi, riducibilità polinomiale e teorema di Cook; algoritmi di approssimazione per problemi di ottimizzazione HP-hard.

Basi di dati

cfu 6 semestre 1

Dr. Vigliano

Introduzione. Algebra relazionale. Calcolo relazionale. Flusso di progetto e visione dei dati. Modello concettuale dei dati. Disegno logico e fisico DB. Forme normali. Query language e implementazioni su MySQL. Simulazione progetto. Realizzazione progetto

Linguaggi di programmazione con laboratorio

cfu 6 semestre 1

Prof. Di Ianni

Parte prima: Caratteristiche dei linguaggi di programmazione. Caratteristiche dei linguaggi di programmazione. Evoluzione dei linguaggi di programmazione. Sintassi: introduzione alle grammatiche formali. Interpretazione e traduzione; il concetto di binding. Variabili: tipo, scope, l-value, r-value. Tipizzazione statica e dinamica. Routines, tecniche di legame dei parametri. Semantica: assiomatica, operativa. Classificazione dei linguaggi rispetto alla struttura run-time. Strutturazione dei dati: tipi built-in e primitivi, dati aggregati, tipi definiti dall'utente e tipi dati astratti. Type systems: program checking statico e dinamico, tipizzazione forte e type checking, compatibilità di tipi, conversioni, tipi e sottotipi, tipi generici, type systems polimorfici. Strutturazione della computazione: espressioni, ed assegnazioni, istruzioni condizionali e iterative, routines ed effetti collaterali. Eccezioni. Computazioni event-driven. Strutturazione del programma: metodi di progetto, concetti di supporto della modularità, caratteristiche del linguaggio per la programmazione in the large, unità generiche. Linguaggi Object-Oriented: concetti di base di programmazione object-oriented, ereditarietà, polimorfismo, binding dinamico delle funzioni. Ereditarietà e type system. Parte seconda: introduzione al linguaggio Java. Introduzione al linguaggio Java: programmi, classi, oggetti, applicazioni; metodi e messaggi; information hiding e modificatori di accesso; editing, compilazione ed esecuzione di una applicazione Java. Struttura di una classe Java: tipi di dato astratti, variabili di istanza e di classe, costruttori, metodi di istanza e di classe. Progetto di applicazioni Java. Array, array multidimensionali. Introduzione all'ereditarietà.

Statistica per la ricerca sperimentale e tecnologica

cfu 6 semestre 1

Dr. Macci

Spazi di probabilità. Probabilità condizionata. Variabili aleatorie discrete. Variabili aleatorie discrete multidimensionali. Variabili aleatorie continue. Legge dei grandi numeri. Teorema limite centrale. Cenno agli intervalli di confidenza

Ricerca operativa Dr. Pacifici	cfu 6	semestre 1
Richiami di algebra lineare e geometria. Problemi e modelli decisionali. Programmazione convessa. Programmazione lineare: poliedri, vertici, soluzioni di base ammissibili. Il metodo del simplesso. Dualità. Ortogonalità delle soluzioni ottime primale/ duale. Formulazione di problemi di decisione come PL. Introduzione alla programmazione lineare intera. Totale unimodularità. Metodo branch and bound per problemi di knapsack binario		
Architettura dei Calcolatori Dr. Gualà (mutuato dal 1 anno)	cfu 6	semestre 2
Laboratorio di Basi di dati Dr. Vigliano	cfu 6	semestre 2
Introduzione. Il processo di ottimizzazione. L'architettura di MySQL. Le transazioni. Le "storage engines". La progettazione fisica di un database . Ottimizzare le istruzioni SQL . Ottimizzare l'uso degli indici. Ottimizzare il motore SQL.		
Metodologie di programmazione con laboratorio Prof. Di Ianni	cfu 6	semestre 2
Prima parte: metodologie di programmazione. Astrazione e decomposizione come metodologia di programmazione. Astrazione per parametrizzazione e per specifica. Tipi di astrazione. Astrazione procedurale. Specifiche. Specifiche per l'astrazione procedurale. Progetto di astrazioni procedurali. Eccezioni. Il meccanismo delle eccezioni in Java. Utilizzo delle eccezioni: riflessione e mascheramento, quando usare le eccezioni, eccezioni checked e unchecked. Astrazione sui dati. Specifiche, implementazione ed utilizzo delle astrazioni sui dati. La funzione di astrazione. L'invariante di rappresentazione. Effetti collaterali benefici. Progetto di astrazioni sui dati: mutabilità, adeguatezza, categorie di operazioni. Astrazione sull'iterazione. Iterazione in Java, Specifica, implementazione ed utilizzo. Invarianti e funzione di astrazione. Progetto di astrazioni sull'iterazione. Gerarchie di tipi. Assegnazione e dispatching. Definizione di una gerarchia di tipi. Tipi eccezione. Classi astratte ed interfacce. Implementazioni multiple. Significato dei sottotipi. Astrazioni polimorfiche. Astrazioni sui dati di tipo polimorfico e suo utilizzo. Flessibilità. Procedure polimorfiche. Seconda parte: linguaggio Java Ereditarietà, classi astratte, interface. Classi interne. Eccezioni. Iteratori. Il sistema I/O di Java (cenni). Finestre ed applet (cenni). RTTI e riflessione.		
Reti di calcolatori Dr. Frasca	cfu 6	semestre 2
Introduzione alle reti e a Internet. Commutazione di circuito e di pacchetto. Reti datagram. Accesso alla rete e mezzi trasmissivi. Ritardi e perdite nelle reti a commutazione di pacchetto. Architettura stratificata. Strato di applicazione. Protocolli. Web e HTTP, FTP, SMTP e MIME, DNS, POP3, IMAP. Applicazioni di rete. Programmazione dei socket con TCP e UDP in Java. Condivisione di file P2P. Strato di trasporto. I protocolli UDP e TCP. Strato di rete. Algoritmi di instradamento link-state e distance vector. Instradamento gerarchico. Protocollo IP. Indirizzamento IPv4 e IPv6. Tecnologia NAT. ICMP. DHCP. Instradamento intra-sistema autonomo: RIP e OSPF. Instradamento inter-sistemi autonomi: BGP. Instradamento multicast. IGMP. Lo strato di collegamento. Adattatori di rete. Tecniche di rilevazione e correzione degli errori. Protocolli di accesso multiplo. LAN. ARP. Ethernet. CSMA/CD. Tecnologie Ethernet. Hub, bridge e switch. TESTI CONSIGLIATI Reti di Calcolatori e Internet - Un approccio top-down (3a edizione) - J.F. Kurose, K.W. Ross - Pearson - Addison W		
Sistemi operativi con laboratorio Dr. Frasca	cfu 6	semestre 2
Storia e classificazione dei sistemi operativi. Struttura dei sistemi operativi. Gestione dei processi. Thread. Mutua esclusione. Modello produttore-consumatore. Semafori. Blocco critico. Gestione della memoria. Gestione dell'I/O. Livello indipendente dai dispositivi. Livello dipendente dai		

dispositivi. Gestione e organizzazione dei dischi. Gestione del file system. I sistemi operativi UNIX e Linux. Architettura di Unix. Interazione con l'utente. Shell. I processi in Unix. Diagramma degli stati. System call per la gestione dei processi. Lo scheduling, la gestione della memoria e il file system di Unix. Protezione in Unix. System call per l'accesso ai file. Interazione tra processi. Sincronizzazione con i segnali. System call per l'uso dei segnali. Comunicazione con pipe. Thread POSIX. La libreria pthread. Sincronizzazione tra thread: mutex e variabili condition.

TESTI CONSIGLIATI

Sistemi operativi - P. Ancillotti, M. Boari, A. Ciampolini, G. Lipari - McGraw-Hill

Sistemi Operativi, VII ed., A. Silberschatz, P. Galvin, G. Gagne. Pearson

Fisica 1

cfu 3 semestre 2

Prof. Goletti

(mutuato dal 1 anno)

3 ANNO

Algoritmi e strutture di dati 2

cfu 6 semestre 1

Prof. Clementi

Verranno illustrate, con un approccio strutturale e mediante numerosi esempi, tecniche generali per la progettazione e l'analisi di algoritmi. Tecnica greedy, programmazione dinamica, riduzioni polinomiali. Problemi NP-completi, algoritmi di approssimazione. Cenni ad algoritmi probabilistici. Si consiglia aver seguito un corso di base di algoritmi.

Testo consigliato: Jon Kleinberg, Eva Tardos. Algorithm Design. Addison Wesley, 2005

Calcolo Numerico

cfu 6 semestre 1

Prof. Bertaccini

Programma sintetico. Algoritmi e risoluzione numerica dei Modelli continui. Aritmetica in precisione finita. Condizionamento e Stabilità. Richiami di Algebra Lineare. Metodi numerici per sistemi lineari. Metodi numerici per equazioni non lineari. Interpolazione ed approssimazione. Approssimazione di integrali definiti. Cenni al trattamento di equazioni differenziali. Cenni all'approssimazione di autovalori e autovettori di matrici

Inglese

cfu 3 semestre 1

Ingegneria del Software

cfu 6 semestre 1

Prof. Intrigila

Caratteristiche dei sistemi software. Sistemi socio-tecnici e sistemi critici. Il ciclo di vita del software: Gestione dei requisiti software. Progettazione. Sviluppo. Verifica e validazione. Metodologie e strumenti per la gestione del ciclo di vita e la produzione del software.

Intelligenza artificiale

cfu 6 semestre 1

Dr. De Luca

Introduzione all'I.A.: definizione di intelligenza, adattività, agente. Tecniche di ricerca: ricerca su grafi, A*. Giochi: minimax, alfa-beta pruning. Rappresentazione della conoscenza: introduzione alla logica matematica, sistemi a frame, reti di conoscenza, RDF e OWL. Dimostrazione automatica di teoremi: completezza e decidibilità programmazione logica, risoluzione SLD, inferenze non monotoniche e logiche modali, introduzione al linguaggio Prolog. Situation calculus. Introduzione ai sistemi esperti: forward e backward reasoning, l'algoritmo Rete, i motori a regole. Problematiche di pianificazione. Introduzione ai sistemi subsimbolici: algoritmi genetici, reti neurali artificiali, a-life. Usi industriali dell'I.A. Il corso utilizzerà il più possibile esempi presi dall'area del **semantic web** e, per la parte simbolica, con esempi pratici di un motore a regole. Avere una visione di insieme dell'intelligenza artificiale, conoscerne le problematiche e le aree principali, saper usare gli strumenti della logica matematica per rappresentare la conoscenza e derivarne inferenze, avere una competenza pratica basilare del linguaggio di programmazione Prolog e di un sistema a regole, nonché una introduzione alle reti neurali e agli algoritmi genetici. Complessivamente lo studente sarà in grado di progettare e implementare semplici componenti intelligenti di un agente.

Linguaggi e Traduttori**cfu 6 semestre 2****Prof. Varricchio**

Linguaggi Formali e Compilatori. Grammatiche e automi. Fasi della compilazione. Analisi Lessicale: token, pattern, lessemi. Il compilatore LEX. Analisi Sintattica: parsing top-down e bottom-up. Parsing predittivo non ricorsivo. Parsing LR: tecniche SLR, LALR, LR canonica. Generazione automatica di compilatori. Il compilatore YACC. Traduzione Guidata dalla Sintassi: definizioni dirette dalla sintassi, schemi di traduzione. Analisi semantica e type checking. Codice intermedio.

Fisica 2**cfu 3 semestre 2****Prof. Goletti**

(mutuato dal 1 anno)

Lab. Sistemi software**cfu 6 semestre 2****Dr. Bei**

Software Development LifeCycle: I processi e le attività a supporto del ciclo di sviluppo del software. Software Project Management: Sizing & Effort Estimation; Planning & Monitoring.

Software Requirement Management. Software Testing: Introduzione al software testing; Static white box testing; Dynamic white box testing; Data based black box testing; Metodologie per il test di applicazioni Web. Tools Open Source

Software Configuration Management, Build & Change Management

Architettura dei Sistemi Software**cfu 6 semestre 2****Prof. Gambosi**

Metodologie, strumenti e processi per la progettazione di architetture software: concetti fondamentali, specifica requisiti funzionali e non funzionali, architetture logiche, progettazione OO, design pattern, sviluppo guidato dai test e refactoring, strumenti UML e UP, metodologie agili. Architetture, Framework e Tecnologie: modelli architetturali di riferimento, architetture e tecnologie J2EE, framework MVC, ORM, IOC.

Analisi Matematica 2**cfu 6 semestre 2****Prof. Liverani****Architettura software per applicazioni in tempo reale ad alta interattività****cfu 5 semestre 2****Dr. Fabricatore**

Introduzione: applicazioni in tempo reale ad alta interattività (ATRAI) e giochi digitali.

ATRAI: caratteristiche fondamentali; Problematiche tipiche delle ATRAI; Stato dell'arte delle ATRAI: i giochi digitali

Architetture software: nozioni fondamentali

Architetture e processi di sviluppo. Caratteristiche di un'architettura software. Principali obiettivi di un'architettura. Strutturazione e tipologie di architetture. Componentizzazione di un sistema. Approcci di distribuzione dei componenti. Stato dell'arte delle architetture stratificate: il modello SunTone

Architetture software in ATRAI e giochi digitali

Il modello da rappresentare: rappresentazione player-centered dell'universo virtuale. Attori ed interazioni in un universo virtuale. Approcci per la componentizzazione di un universo virtuale. Attori. Componenti di servizi. Principi di *agile modeling & programming*. Introduzione a SCRUM: pragmatismo e project management per progetti ad alto rischio. Approcci di distribuzione dei componenti. Introduzione all'agile modelling. Introduzione al re factoring. Caso di studio: Tetris

Problematiche di distribuzione ed interoperatività. Comunicazione tra componenti (accoppiamento nella comunicazione). Persistenza di stato. Informazioni di stato. Patterns per l'interoperatività tra componenti distribuite. Interoperatività e performance. Qualità sistemiche di primario interesse. Sicurezza. Performance. Disponibilità. Design e l'implementazione di architetture: i patterns. Natura, utilizzo e livelli di applicabilità di un pattern. Architectural patterns. Software patterns

Modellazione architettonica di un gioco digitale

Agenti. Strutturazione e design. Controllo del comportamento. Gestione e controllo di status.

Controllo interno del ciclo di vita. Software & architectural patterns. Game Mechanics. Strutturazione e design. Controllo del comportamento. Controllo interno del ciclo di vita. Software & architectural patterns

Servizi e componenti di comunicazione. Sistemi di comunicazione diretta. Sistemi di comunicazione ad eventi. Software & architectural patterns

Servizi e componenti di rendering. 2D rendering. Scene graph. Software & architectural patterns

Servizi e componenti di world management. Area management. Sistemi di regole. Gestione di persistenza. Monitoraggio e consultazione dello status del mondo. Gestione del ciclo di vita degli attori. Software & architectural patterns

Interfacce grafiche. Controllo dello status dell'universo. Comunicazione diretta ed indiretta con gli attori. Software & architectural patterns