

# INFORMATICA

## GUIDA DELLO

### STUDENTE AA. 2012-2013

**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, così come si parla del periodo attuale come di una "terza rivoluzione industriale", basata sull'elettronica e sull'informatica, dopo la prima, basata sulla macchina a vapore, e la seconda, basata sull'elettricità.

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 triennale in Informatica 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.

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 magistrale, oppure entrare subito nel mondo del lavoro.

### **Ordinamento degli Studi**

La descrizione dei contenuti didattici dei corsi di laurea fa riferimento ai crediti formativi universitari (cfu) come unità di misura dell'impegno stimato richiesto per una adeguata acquisizione dei contenuti dei vari insegnamenti. Ogni credito vale, convenzionalmente, 25 ore di lavoro (comprendendo le ore di lezione, di esercitazione e il lavoro individuale). Inoltre, è stato stabilito che 1 credito corrisponda al lavoro necessario per seguire e comprendere 8 ore di lezione oppure 12 di esercitazione. Per potersi laureare lo studente dovrà maturare almeno 180 crediti nell'ambito delle varie attività didattiche (compresa la prova finale). Per una uniforme ripartizione del carico di lavoro nell'arco dei tre anni, il numero di crediti associati agli insegnamenti di ognuno degli anni di corso è pari a (circa) 60.

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 Studio (CCS) e pubblicati sul sito web del corso di laurea.

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 CCS ed il superamento dei relativi esami. Lo studente effettua tale scelta presentando al CCS un piano di studio, secondo le modalità previste. L'acquisizione di tali crediti mediante differenti meccanismi verrà sempre valutata dal CCS 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 CCS. Tali opzioni possono prevedere: (1) frequenza di ulteriori insegnamenti e superamento delle relative prove d'esame; (2) frequenza di corsi brevi o cicli di seminari specificatamente organizzati ai fini dell'acquisizione di crediti di tipo F: il CCS darà preventiva informazione di tali iniziative, precisando i crediti relativi e le modalità per la loro acquisizione; (3) effettuazione di attività supplementari, assegnate su base individuale, nell'ambito di insegnamenti o della prova finale; (4) attività svolte nell'ambito di stage concordati preventivamente con il CCS.

Sul sito web del corso di laurea 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 studenti interessati ad iscriversi al corso di laurea in Informatica devono sostenere una prova di valutazione per la verifica delle conoscenze matematiche di base propedeutiche ai contenuti trattati nel corso di laurea. Tale prova consisterà nella risposta ad un insieme di quiz a risposta multipla, e si terrà una prima volta nei giorni dal 4 al 7 settembre 2012 nelle aule della Facoltà di Scienze MFN. E' necessario che lo studente si prenoti su internet attraverso il sito della Facoltà di Scienze ([www.scienze.uniroma2.it](http://www.scienze.uniroma2.it)) – seguendo il link: Area Studenti poi Test d'ingresso. Gli studenti che non dovessero sostenere o superare la prova hanno la possibilità di colmare le proprie lacune seguendo un apposito corso intensivo di Matematica di base, denominato Matematica 0, che si terrà nel mese di settembre. Nei giorni 2 e 3 ottobre 2012 si terrà una seconda prova di recupero che interesserà gli studenti che non hanno partecipato alla prima e/o gli studenti che non l'abbiano superata. Agli studenti che non abbiano superato

né il primo né il secondo test di valutazione delle conoscenze in ingresso (previsto dal DM 270/04 per i corsi di laurea che non hanno il numero programmato) o che non vi abbiano partecipato, verrà assegnato come debito formativo aggiuntivo "l'obbligo di superare come primi due esami gli insegnamenti di Analisi Matematica e Matematica Discreta". Ai sensi del DM 270/04 gli obblighi formativi aggiuntivi vanno colmati durante il primo anno di corso.

### **Orientamento**

Oltre ai numeri di telefono ed al sito internet indicati, per chi desidera informazioni sul Corso di Laurea in Informatica, così come pure per gli altri corsi di Laurea della Facoltà di Scienze, verrà organizzato un servizio di accoglienza nel periodo delle iscrizioni.

Sarà possibile in particolare avere informazioni sulle modalità di iscrizione, sul contenuto dei corsi e dialogare con gli studenti dei Corsi di Laurea. Per informazioni sulla didattica, lo studente si può rivolgere alla segreteria del Corso di Laurea, Sig.ra Laura Filippetti, tel. 06 72594839, presso il Dipartimento di Matematica. Le informazioni sono comunque riportate nel sito web del corso di Laurea [www.informatica.uniroma2.it](http://www.informatica.uniroma2.it). Ulteriori informazioni si possono anche ottenere per posta elettronica all'indirizzo [infordid@mat.uniroma2.it](mailto:infordid@mat.uniroma2.it).

### **Esami**

I corsi del primo semestre prevedono due appelli di esame nella sessione estiva anticipata (febbraio) un appello nella sessione estiva (giugno-luglio) e uno in quella autunnale (settembre). I corsi del secondo semestre e quelli che si svolgono nel corso di entrambi i semestri prevedono due appelli d'esame nella sessione estiva, uno in quella autunnale e uno a febbraio.

### **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 CCS, 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 del corso di laurea.

Il voto finale viene determinato a partire dalla media dei voti riportati dallo studente sulle attività formative che prevedono l'assegnazione di un voto. La media è calcolata pesando i voti dei vari insegnamenti con il numero dei relativi cfu: a tale media, la Commissione per la prova finale aggiunge un punteggio determinato dal giudizio sul lavoro svolto. Inoltre, agli studenti che svolgono la prova finale in corso o nel loro quarto anno di studi viene attribuito un ulteriore premio nel punteggio. Sulla base di tutto ciò, la Commissione esprime un voto in centodecimali:

agli studenti che raggiungono il punteggio di 111 può essere attribuita, sulla base della valutazione della prova finale, la lode.

### **Tutorato**

Ad ogni studente immatricolato viene assegnato, entro il mese di Dicembre, un docente tutor che potrà essere consultato, per consigli e suggerimenti generali in merito all'andamento delle attività di studio. Tutti i docenti dei corsi hanno un orario di ricevimento settimanale per eventuali chiarimenti da parte degli studenti sulla materia insegnata.

### **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. Il Consiglio di Corso di Laurea propone un curriculum di studi che prevede le modalità di assegnazione di tutti i crediti formativi, ad eccezione di quelli di tipo D: tale curriculum viene pubblicato sul sito web del corso di laurea. Lo studente che intende adottare tale curriculum ha il solo obbligo di specificare le modalità di acquisizione dei crediti di tipo D. Piani di studi diversi sono comunque soggetti a procedura di valutazione da parte del CCS ai fini dell'accettazione del piano di studio stesso.

### **Trasferimenti**

Gli studenti si trasferiscono al Corso di Laurea in Informatica provenendo da altri Corsi di Studi possono chiedere il riconoscimento dei crediti relativi ad esami sostenuti nel corso di studi d'origine. Il Consiglio di Corso di Studi valuterà di volta in volta le singole richieste. Si precisa che i trasferimenti non possono avvenire su corsi disattivati. Sul sito web del corso di studio [www.informatica.uniroma2.it](http://www.informatica.uniroma2.it) nella sezione "trasferimenti" si possono leggere le istruzioni

per ottenere un parere preventivo su eventuali convalide di esami sostenuti in precedenti corsi di laurea di provenienza.

## **Ordinamento degli Studi - Laurea Triennale**

### **Piano delle attività formative**

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

#### **Attività di base**

##### **I ANNO**

Programmazione dei calcolatori con laboratorio	12 crediti	1° e 2° Semestre
Architettura e sistemi di elaborazione	6 crediti	1° Semestre
Analisi Matematica	9 crediti	1° Semestre
Matematica discreta	9 crediti	1° Semestre
Fondamenti di informatica	6 crediti	2° Semestre
Fisica	6 crediti	2° Semestre
Geometria ed algebra	6 crediti	2° Semestre
Inglese	3 crediti	2° Semestre

##### **2 ANNO**

Algoritmi e strutture di dati	12 crediti	1° e 2° Semestre
Basi di dati	12 crediti	1° e 2° Semestre
Programmazione a oggetti	12 crediti	1° e 2° Semestre
Calcolo delle probabilità	6 crediti	1° Semestre
Ricerca operativa	6 crediti	2° Semestre
Sistemi operativi e reti di calcolatori	12 crediti	1° e 2° Semestre

##### **3 ANNO**

Calcolo Numerico	6 crediti	1° Semestre
Ingegneria del Software	12 crediti	1° e 2° Semestre
Intelligenza artificiale	12 crediti	1° e 2° Semestre
Informatica teorica	6 crediti	2° Semestre
Algoritmi e strutture di dati 2	6 crediti	2° Semestre

#### **Corsi a scelta**

Algoritmi e strutture dati 2 (6CFU)

Modelli e linguaggi di simulazione (6CFU)

Laboratorio avanzato di Informatica (6CFU)

Logica per la programmazione e dimostrazione automatica (6CFU)

Programmazione web (6CFU)

## **Calendario 2011/2012**

I corsi hanno durata semestrale.

## **Informazioni**

Presso la Presidenza della Facoltà di Scienze Matematiche Fisiche e Naturali, nell'edificio SOGENE in via della Ricerca Scientifica si trova la bacheca in cui vengono affissi gli avvisi e tutte le informazioni pratiche (orari dei corsi, calendario degli esami, orari di ricevimento dei docenti). Per informazioni sulla didattica, lo studente si può rivolgere alla segreteria del Corso di Laurea, Sig.ra Laurea Filippetti, tel. 06 72594839, presso il Dipartimento di Matematica. Le informazioni sono comunque riportate nel sito web del corso di laurea ([www.informatica.uniroma2.it](http://www.informatica.uniroma2.it)).

## **Programmi dei Corsi**

### **I ANNO**

#### **Programmazione dei calcolatori con laboratorio (INF/01)**

**cfu 12**

Introduzione alla programmazione: Il modello di calcolo. Storia e caratterizzazione dei linguaggi di programmazione. Sintassi e semantica di un linguaggio. Tipi di dati e variabili. Programmazione strutturata: espressioni, istruzioni per il controllo del flusso. Dati strutturati: array, stringhe, puntatori. Procedure e funzioni. Metodi di passaggio dei parametri. Contesto di esecuzione. Sistema run-time. Allocazione dinamica della memoria, gestione dell'I/O. Il linguaggio C come esempio di linguaggio strutturato. Algoritmi fondamentali: principali algoritmi di ordinamento, introduzione e utilizzo delle strutture dati di base e implementazione code, stack, liste, alberi.

#### **Architettura dei sistemi di elaborazione (INF/01)**

**cfu 6**

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. Sarà trattata la struttura interna dei calcolatori (CPU, Memoria, dischi e altri dispositivi).

#### **Analisi Matematica (MAT/05)**

**cfu 9**

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. Funzioni continue. 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 di 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 (MAT/03)**

**cfu 9**

Relazioni d'ordine e di equivalenza. Insiemi quoziente. Cardinalità di insiemi. Combinatoria. Il principio di inclusione-esclusione. Proprietà 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 irriducibilità su vari campi. Campi finiti e loro costruzione. Le fondamentali strutture algebriche: gruppoidi, semigruppoidi, monoidi, gruppi, anelli, campi. Il gruppo simmetrico. Cenni di logica: calcolo proposizionale, calcolo dei predicati.

**Fondamenti di informatica (INF/01)****cfu 6**

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. Proprietà 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. Proprietà 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 (FIS/01)****cfu 6**

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 (MAT/03)****cfu 6**

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 autovettori autovalori 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 (INF/01)****cfu 12**

Il corso offre un'introduzione allo studio degli algoritmi e delle strutture dati e ha come obiettivo l'acquisizione delle metodologie e delle tecniche utili per la progettazione e l'analisi di algoritmi e strutture dati efficienti.

Il corso è diviso in due moduli. Gli argomenti del primo modulo sono:

Analisi degli algoritmi e complessità asintotica; Tecniche per l'analisi di algoritmi ricorsivi; Tecniche di progettazione di algoritmi (divide-et-impera, programmazione dinamica); Algoritmi di ordinamento e ricerca; Strutture dati (array, liste, alberi binari di ricerca bilanciati, code con priorità).

Del secondo modulo fanno parte i seguenti argomenti: I grafi e le loro proprietà; Algoritmi di visita di grafi non pesati e loro applicazioni; Trattabilità ed intrattabilità dei problemi; Il problema dei cammini minimi su grafi pesati (algoritmi di Dijkstra, Bellmann-Ford e Floyd-Warshall); Il problema del minimo albero coprente su grafi non diretti pesati (algoritmi di Prim e Kruskal); Problemi su grafi diretti pesati (arborescenza minima, minimo sotto-grafo fortemente connesso).

**Basi di dati (INF/01)****cfu 6**

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. 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.

### **Programmazione a oggetti (INF/01)**

**cfu 12**

Caratteristiche dei linguaggi di programmazione: Evoluzione dei linguaggi di programmazione. Sintassi: introduzione alle grammatiche formali. Interpretazione e traduzione; il concetto di binding. Variabili. Tipizzazione statica e dinamica. Routines, tecniche di legame dei parametri. Semantica: assiomatica, operativa. Classificazione dei linguaggi rispetto alla struttura run-time. Strutturazione dei dati. Type systems e type checking. 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.

Metodologie di programmazione: Astrazione e decomposizione come metodologia di programmazione. Astrazione per parametrizzazione e per specifica. Tipi di astrazione. Astrazione procedurale. Eccezioni. Utilizzo delle eccezioni. Specifiche, implementazione ed utilizzo delle astrazioni sui dati. Astrazione sull'iterazione.

Gerarchie di tipi. Classi astratte ed interfacce. Astrazioni polimorfiche. Procedure polimorfiche.

Il 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. Ereditarietà, classi astratte, interfacce. Classi interne. Eccezioni. Iteratori.

### **Calcolo delle probabilità e statistica (MAT/06)**

**cfu 6**

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 (MAT/09)**

**cfu 6**

Richiami di algebra lineare e geometria. Programmazione lineare. Il metodo del simplesso. Soluzione numerica del metodo del simplesso. Tecniche di base per l'ottimizzazione numerica.

### **Sistemi operativi e reti di calcolatori**

**cfu 12**

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. 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.

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. Reti wireless.

## **3 ANNO**

### **Algoritmi e strutture di dati 2 (INF/01)**

**cfu 6**

Studio delle principali tecniche algoritmiche per problemi combinatorici provenienti da diversi ambienti applicativi (reti di comunicazione, scheduling di risorse, riconoscimento di sequenze del

DNA, compressione dati). Tale studio è svolto mediante un insieme omogeneo di esempi globali dell'approccio matematico-informatico per la risoluzione di problemi (Problem Solving):  
APPLICAZIONE dal MONDO REALE ----> MODELLO MATEMATICO DEL PROBLEMA ----> SOLUZIONE ALGORITMICA ----> ANALISI MATEMATICA DELLA CORRETTEZZA E DELL'EFFICIENZA DELLA SOLUZIONE PROPOSTA

Questi esempi globali sono organizzati temporalmente in classi omogenee determinate dalla tecnica algoritmica utilizzata. Pertanto, lo scopo è quello di individuare gli aspetti strutturali tipici di una tecnica algoritmica riscoprendoli nella risoluzione di diversi problemi di varia natura applicativa.

Le tecniche fondamentali sono TRE:

Il Metodo Goloso (GREEDY)

La Programmazione Dinamica

Le Riduzioni Polinomiali

Inoltre, tempo permettendo, un ciclo di alcune lezioni finali saranno dedicate agli algoritmi randomizzati.

### **Calcolo Numerico (MAT/08)**

**cfu 6**

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.

### **Ingegneria del Software (INF/01)**

**cfu 12**

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. Metodologie, strumenti e processi per la progettazione di architetture software: concetti fondamentali, specifica requisiti funzionali e nonfunzionali, 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.

### **Intelligenza artificiale (INF/01)**

**cfu 12**

Analizzare le metodologie, i modelli e le implementazioni dei sistemi per la gestione delle informazioni e della conoscenza, cioè delle diverse astrazioni riguardo alla manipolazioni di grandi quantità di dati in un sistema informatico.

Attenzione alle problematiche della gestione della conoscenza nel contesto del Semantic Web. Rappresentazione della Conoscenza e Ragionamento automatico. Linguaggi per la gestione della conoscenza ontologica.

### **Informatica Teorica (INF/01)**

**cfu 6**

Nella prima parte del corso verranno introdotti i concetti di calcolabilità di funzioni e di decidibilità ed accettabilità di linguaggi, e verranno analizzate le relazioni che intercorrono fra tali concetti.

Successivamente, nella seconda parte del corso, si studierà la complessità intrinseca dei problemi: dopo avere introdotto le misure dinamiche di complessità, verranno definite ed analizzate le classi P, NP, e le classi oltre NP, nonché il concetto di completezza per una classe. Infine, verranno introdotti i problemi di ottimizzazione e le corrispondenti classi NPO, PO ed APX.