



Nuovo ordinamento del

Corso di laurea in Informatica

Classe L-31 (Scienze e tecnologie informatiche)
ex D.M. 270/2004

| | |
|---|----|
| 1) Introduzione | 2 |
| 2) Il nuovo corso di laurea in Informatica..... | 2 |
| 2.1) Obiettivi formativi e sbocchi professionali..... | 2 |
| 2.2) Conoscenze richieste per l'accesso | 3 |
| 3) Quadro generale dell'offerta formativa | 4 |
| 3.1) Tirocinio formativo e prova finale..... | 5 |
| 4) Articolazione del corso di laurea in curricula | 6 |
| 4.1) Curriculum metodologico | 6 |
| 4.2) Curriculum tecnologico | 7 |
| 4.3) Insegnamenti complementari..... | 8 |
| 5) Contenuti degli insegnamenti | 9 |
| 5.1) Insegnamenti di primo anno | 9 |
| 5.2) Insegnamenti di secondo anno..... | 10 |
| 5.3) Insegnamenti di terzo anno | 12 |
| 5.3.1) Insegnamenti metodologici | 12 |
| 5.3.2) Insegnamenti tecnologici..... | 13 |
| 5.3.3) Insegnamenti complementari | 14 |



1) Introduzione

Questo documento descrive le caratteristiche e la struttura del nuovo corso di laurea in Informatica, il cui ordinamento è progettato in conformità al D.M. 270 del 2004, che sostituirà i corsi di laurea triennale in Informatica e Tecnologie informatiche previsti dall'attuale ordinamento (che è invece conforme al D.M. 509 del 1999). L'attivazione del nuovo corso di laurea avverrà nell'a.a. 2009-2010. Saranno attivati contemporaneamente il primo ed il secondo anno di corso.

2) Il nuovo corso di laurea in Informatica

2.1) Obiettivi formativi e sbocchi professionali

Il corso di laurea in Informatica ha l'obiettivo di formare figure professionali capaci di affrontare con successo le sfide costituite dalle crescenti esigenze della società dell'informazione. I laureati in Informatica saranno dotati di solida preparazione culturale di base, che permetterà loro di mantenersi al passo col progredire delle tecnologie, e di preparazione tecnica che consentirà loro un rapido inserimento professionale nel settore delle tecnologie dell'informazione e della comunicazione. Inoltre, saranno in grado di accedere ai livelli di studio universitario successivi al primo, nel settore dell'informatica.

La preparazione culturale di base permetterà ai laureati in Informatica di avere:

- familiarità col metodo scientifico di indagine;
- capacità di comprendere ed utilizzare strumenti matematici di supporto;
- conoscenze metodologiche e competenze di base in un ampio spettro di settori delle scienze e delle tecnologie dell'informazione e della comunicazione;
- familiarità con almeno una lingua dell'Unione Europea.

Grazie a tali solide basi teoriche, metodologiche e tecnologiche, i laureati in Informatica saranno in grado di:

- comprendere l'evoluzione tecnologica ed adeguarsi al progredire delle discipline informatiche, anche attraverso la consultazione di documentazione avanzata di tipo scientifico e tecnologico;
- possedere buone capacità ed autonomia nella costruzione di modelli indispensabili per la comprensione e la formalizzazione di problemi complessi;
- operare nella progettazione, sviluppo e gestione di sistemi software (anche in ambiente di rete), di sistemi informativi, di reti di elaboratori, di soluzioni per la sicurezza dei sistemi di calcolo;
- fornire supporto tecnologico ad utilizzatori di sistemi informatici;
- inserirsi rapidamente ed efficacemente in una realtà lavorativa, operando sia in gruppo che in autonomia;



- comunicare ed argomentare le proprie idee in merito ai problemi affrontati ed alle soluzioni proposte, tanto ad interlocutori specialisti che non specialisti;
- accedere al successivo livello di studi costituito dalle Lauree magistrali della classe LM-18 (Informatica).

I laureati in Informatica saranno in grado di svolgere attività professionale sia in aziende produttrici che in aziende utilizzatrici di sistemi informatici, operanti tanto nel settore pubblico che privato, nei seguenti ambiti occupazionali:

- progettazione, organizzazione, manutenzione anche evolutiva e gestione di sistemi software, software applicativo, basi di dati, sistemi informativi, applicazioni di rete client-server, siti Web;
- progettazione, organizzazione, manutenzione anche evolutiva delle componenti di supporto alla sicurezza ed alla affidabilità dei sistemi informatici.

I laureati in Informatica potranno accedere alle professioni della categoria ISTAT dei Tecnici informatici e ad alcune delle professioni della categoria ISTAT degli Informatici e telematici, e più specificamente alle professioni di:

- Analisti e progettisti di software applicativi e di sistema;
- Analisti di sistema.

Inoltre, la loro formazione potrà essere completata attraverso un corso di laurea di secondo livello di area Informatica in modo da dare loro accesso alle professioni specialistiche presenti nella categoria ISTAT degli Informatici e telematici.

I laureati in Informatica possono accedere all'esame di stato per l'iscrizione all'Albo degli ingegneri dell'informazione (Albo professionale - sezione B degli ingegneri junior - settore dell'informazione).

2.2) Conoscenze richieste per l'accesso

Per essere ammessi al corso di laurea occorre essere in possesso di un diploma di scuola secondaria superiore, ovvero di altro titolo di studio conseguito all'estero, riconosciuto idoneo.

Per affrontare con successo il percorso di studio sono necessarie le conoscenze elementari di fisica e matematica che sono fornite di norma dal ciclo di studi secondario superiore. E' richiesta altresì capacità logica e di comprensione dei testi scritti e del discorso, nonché padronanza di espressione attraverso la scrittura.

L'accesso al corso di laurea in Informatica è libero. Gli studenti devono sostenere una prova di ingresso per la verifica delle conoscenze. I risultati del test servono inoltre agli studenti come importante valutazione delle proprie capacità e attitudini, in vista dell'eventuale prosecuzione degli studi informatici.

Per gli studenti per i quali la verifica non risulterà positiva sono previste forme di recupero nell'ambito dei corsi del primo anno. Per coloro che, pur superando il test, non ottengono risultati brillanti sono offerti dei precorsi di matematica prima dell'inizio delle lezioni.



3) Quadro generale dell'offerta formativa

Il primo anno prevede i seguenti insegnamenti, obbligatori per tutti:

| Insegnamento | Sem. | Crediti | SSD | Ambito |
|-------------------------------------|------|---------|----------|-----------|
| Calcolo differenziale | I | 6 | Mat/05 | Di base |
| Calcolo integrale | II | 6 | Mat/05 | Di base |
| Progettazione di sistemi digitali | I | 6 | Inf/01 | Di base |
| Architettura degli elaboratori | II | 6 | Inf/01 | Di base |
| Metodi matematici per l'informatica | I | 6 | Inf/01 | Caratt. |
| Introduzione agli algoritmi | II | 6 | Inf/01 | Di base |
| Fondamenti di programmazione | I | 9 | Inf/01 | Di base |
| Metodologie di programmazione | II | 9 | Inf/01 | Caratt. |
| Inglese | | 6 | L-LIN-12 | L. stran. |
| Totali | | 60 | | |

Il secondo anno prevede i seguenti insegnamenti, obbligatori per tutti:

| Insegnamento | Sem. | Crediti | SSD | Ambito |
|----------------------------|------|---------|--------|---------|
| Algebra | I | 9 | Mat/02 | Affine |
| Basi di dati | II | 12 | Inf/01 | Caratt. |
| Calcolo delle probabilità | II | 9 | Mat/06 | Affine |
| Progettazione di algoritmi | I | 9 | Inf/01 | Caratt. |
| Reti di elaboratori | II | 9 | Inf/01 | Caratt. |
| Sistemi operativi | I | 12 | Inf/01 | Caratt. |
| Totali | | 60 | | |



Il terzo anno è strutturato come segue.

| Insegnamento | Sem. | Crediti | SSD | Ambito |
|--|------|---------|--------|---------|
| Ingegneria del software (obbligatorio) | I | 6 | Inf/01 | Caratt. |
| Insegnamento curriculare | | 6 | | |
| Insegnamento curriculare | | 6 | | |
| Insegnamento curriculare | | 6 | | |
| Insegnamento curriculare | | 6 | | |
| A scelta dello studente | | 12 | | |
| Tirocinio | II | 12 | | |
| Prova finale | II | 6 | | |
| Totali | | 60 | | |

3.1) Tirocinio formativo e prova finale

Per tutti gli studenti è previsto al terzo anno un tirocinio formativo. Esso è svolto sotto la guida di un responsabile e può essere esterno (svolto presso aziende o enti esterni) o interno (svolto nell'ambito del corso di laurea). In entrambi i casi il tirocinio ha una durata di circa tre mesi e prevede che allo studente sia proposto un problema del mondo reale, che egli deve risolvere attraverso l'elaborazione di un progetto sviluppato con un approccio professionale.

La prova finale consiste nella stesura, nella presentazione e nella discussione di una relazione scritta, elaborata autonomamente dallo studente, che documenti in modo organico e dettagliato il problema affrontato nell'ambito del tirocinio formativo e tutte le attività compiute per pervenire alla soluzione. La discussione si svolge di fronte alla Commissione di laurea che, sulla base della carriera dello studente e della valutazione della relazione, stabilisce il voto di laurea.



4) Articolazione del corso di laurea in curricula

Il corso di laurea è articolato in due curricula, ciascuno dei quali ha una specifica finalità. Sono previsti un curriculum *metodologico* ed un curriculum *tecnologico*. Gli studenti scelgono il curriculum e formulano il proprio piano di studio all'inizio del terzo anno.

4.1) Curriculum metodologico

Il curriculum metodologico è consigliato a chi vuole ottenere la formazione più idonea alla prosecuzione degli studi nella laurea magistrale in Informatica.

Esso:

- *prescrive* che i quattro insegnamenti curriculari del terzo anno vengano scelti nella tabella degli insegnamenti metodologici;
- *suggerisce* che i 12 crediti a scelta dello studente vengano utilizzati per scegliere i restanti due insegnamenti della tabella degli insegnamenti metodologici.

Tabella degli insegnamenti metodologici

| Insegnamento | Crediti | SSD | Ambito |
|------------------------------|---------|--------|---------|
| Combinatoria per Informatica | 6 | Inf/01 | Affine |
| Fisica | 6 | Fis/01 | Affine |
| Intelligenza artificiale | 6 | Inf/01 | Caratt. |
| Modelli e ottimizzazione | 6 | Inf/01 | Caratt. |
| Paradigmi di programmazione | 6 | Inf/01 | Caratt. |
| Sicurezza | 6 | Inf/01 | Caratt. |

Obiettivi formativi

Il curriculum metodologico offre la possibilità di approfondire conoscenze scientifiche di base della fisica e della matematica discreta per l'informatica di oggi e del futuro. Inoltre consente allo studente di conoscere i principi, i metodi, i problemi e gli approcci risolutivi propri di alcune delle grandi aree tematiche dell'informatica non trattate nel biennio: l'intelligenza artificiale, i linguaggi di programmazione, l'ottimizzazione combinatoria, la sicurezza.

Gli obiettivi formativi specifici del curriculum metodologico sono:

- fornire gli strumenti metodologici, scientifici e tecnologici necessari per affrontare con successo il rapido evolversi delle tecnologie informatiche e delle loro applicazioni;



- fornire le competenze di base e fondamentali per comprendere, saper utilizzare e poter applicare le tecnologie informatiche attuali e future;
- approfondire le nozioni, gli strumenti, i metodi necessari per proseguire gli studi nella laurea magistrale e sapersi orientare consapevolmente verso aree di interesse specialistico dell'Informatica.

Sbocchi professionali

Il curriculum metodologico è progettato per fornire le conoscenze teorico-metodologiche più indicate per proseguire con successo gli studi nell'ambito della laurea magistrale in Informatica. Queste conoscenze, unite a quelle applicative di base, permetteranno ai laureati che abbiano scelto questo percorso e che poi si inseriscano in un ambito lavorativo di apprendere rapidamente le tecnologie necessarie per poter operare proficuamente nel contesto aziendale.

4.2) Curriculum tecnologico

Il curriculum tecnologico è consigliato a chi vuole ottenere una formazione idonea ad un proficuo inserimento in realtà produttive.

Esso:

- *prescrive* che due degli insegnamenti curriculari del terzo anno vengano scelti nella tabella degli insegnamenti metodologici;
- *prescrive* che gli altri due insegnamenti curriculari del terzo anno vengano scelti nella tabella degli insegnamenti tecnologici;
- *suggerisce* che i 12 crediti a scelta dello studente vengano utilizzati per scegliere i restanti due insegnamenti della tabella degli insegnamenti tecnologici.

Tabella degli insegnamenti tecnologici

| Insegnamento | Crediti | SSD | Ambito |
|---------------------------|---------|--------|---------|
| Interazione uomo macchina | 6 | Inf/01 | Caratt. |
| Programmazione di sistema | 6 | Inf/01 | Caratt. |
| Programmazione per il Web | 6 | Inf/01 | Caratt. |
| Sistemi di basi di dati | 6 | Inf/01 | Caratt. |

La tabella potrà essere ampliata, sulla base delle effettive disponibilità di docenza.

Obiettivi formativi

Il curriculum tecnologico è finalizzato alla formazione di un professionista dotato di solide conoscenze di base e di preparazione tecnica adeguata ad un rapido ed efficace inserimento in uno dei settori lavorativi collegati agli insegnamenti tecnologici scelti nell'ambito del curriculum.



Gli approfondimenti del curriculum tecnologico mirano a fornire le seguenti capacità e competenze professionali:

- conoscenza dei principi e delle metodologie per la progettazione, lo sviluppo, l'ottimizzazione e la valutazione di una base di dati;
- conoscenza delle problematiche fondamentali, e delle relative soluzioni, che si incontrano nella progettazione di interfacce utente da utilizzarsi sia su elaboratori fissi che portatili, nonché le principali tecniche per la valutazione di usabilità;
- conoscenza dei costrutti linguistici principali e delle tecnologie più diffuse, con particolare riferimento ai moderni linguaggi di programmazione orientati ad oggetti, per lo sviluppo di contenuti e servizi per il Web;
- conoscenza dei metodi e degli strumenti per la programmazione sequenziale e concorrente, per la comunicazione tra processi e, più in generale, per la programmazione di sistema;
- esperienza di lavoro in gruppo.

Sbocchi professionali

Il curriculum è finalizzato alla formazione di un professionista capace di inserirsi efficacemente in aziende operanti sia nel settore pubblico che in quello privato, contribuendo a due o più delle seguenti attività professionali:

- progettazione, gestione e manutenzione di basi di dati;
- progettazione e valutazione di interfacce utente;
- progettazione, gestione e manutenzione di applicazioni e servizi per il Web;
- progettazione, gestione e manutenzione di software di sistema.

4.3) Insegnamenti complementari

Oltre agli insegnamenti curriculari metodologici e tecnologici, vengono offerti alcuni insegnamenti complementari, quindi non curriculari, volti ad offrire agli studenti che lo desiderino la possibilità di completare la propria preparazione di tipo interdisciplinare. Gli insegnamenti complementari possono essere scelti nell'ambito dei 12 crediti a scelta dello studente.

Tabella degli insegnamenti complementari

| Insegnamento | Crediti | SSD |
|-----------------------|----------------|------------|
| Calcolo numerico | 6 | Mat/08 |
| Economia aziendale | 6 | SECS-P/07 |
| Informatica e diritto | 6 | Ius/20 |

La tabella potrà essere ampliata, sulla base delle effettive disponibilità di docenza.



5) Contenuti degli insegnamenti

5.1) Insegnamenti di primo anno

Calcolo differenziale

E' un insegnamento introduttivo di analisi matematica. Si studiano le successioni e i loro limiti, le funzioni reali (limiti, funzioni continue, teorema degli zeri, continuità uniforme; funzioni monotone) e le serie numeriche, le serie a termini positivi e a segni alterni, inoltre derivate, massimi e minimi, funzioni convesse e la formula di Taylor.

Calcolo integrale

E' fortemente integrato con l'insegnamento di Calcolo differenziale che lo presuppone. Vi si studiano serie e successioni di funzioni; serie di potenze e serie di Taylor, gli integrali semplici e il teorema fondamentale del calcolo. Si introducono il calcolo differenziale ed integrale per funzioni di più variabili, le equazioni differenziali del primo ordine e lineari del secondo ordine con coefficienti costanti.

Progettazione di sistemi digitali

Tratta argomenti preliminari alla descrizione dell'organizzazione di un calcolatore: rappresentazione dell'informazione, progettazione di reti combinatorie e sequenziali, registri e loro interconnessione. L'insegnamento fornisce gli strumenti e le competenze necessari alla comprensione del funzionamento e dei problemi di progetto di un calcolatore, argomenti che vengono trattati nell'insegnamento di Architettura degli elaboratori.

Architettura degli elaboratori

L'insegnamento di Architettura degli elaboratori descrive l'organizzazione e l'architettura di un calcolatore, senza far riferimento ad una specifica macchina, ma fornendo una descrizione funzionale dei componenti dell'architettura di Von Neumann su cui le attuali macchine sono modellate. Si descrivono anche alcune tecniche di miglioramento delle prestazioni come: CU microprogrammata, pipeline delle istruzioni, memoria cache, memoria virtuale.

Metodi matematici per l'informatica

L'insegnamento si propone di far familiarizzare lo studente con la formalizzazione matematica e con il metodo di deduzione. Verranno illustrati i risultati di base della teoria degli insiemi la struttura dei numeri naturali ed il principio di induzione. Seguirà un'introduzione alla logica matematica mettendo in evidenza l'importanza del calcolo logico per l'informatica.

Introduzione agli algoritmi

Analisi della complessità di algoritmi ricorsivi ed iterativi. Algoritmi fondamentali di ordinamento, fusione, ricerca e analisi della loro efficienza. Strutture dati di base (pile, code, alberi, code con priorità): algoritmi fondamentali e analisi della loro efficienza. Strutture dati avanzate (tabelle hash, alberi di ricerca): algoritmi fondamentali e analisi della loro efficienza.

Fondamenti di programmazione

L'obiettivo dell'insegnamento è fornire le basi della programmazione dei calcolatori: nozioni introduttive. Flusso del controllo. Scomposizione di un programma: funzioni, procedure, parametri, passaggio dei parametri. Programmazione ricorsiva su dati scalari. Iterazione. Vettori, stringhe, puntatori e aritmetica dei puntatori in C. Ricorsione su vettori. Gestione della memoria



in C. Strutture dinamiche di dati: liste. Attività di laboratorio: familiarizzazione con l'ambiente di programmazione, costrutti del linguaggio C, implementazione di programmi visti e loro varianti, puntatori e gestione della memoria.

Metodologie di programmazione

E' un insegnamento integrato con Fondamenti di programmazione per portare lo studente ad ottenere programmi di buona qualità: Tipi astratti di dato. Concetti introduttivi della programmazione ad oggetti. Introduzione alle metodologie di disegno. Implementazione di strutture dinamiche di dati lineari e non lineari. Ricorsione base ed avanzata (generazione di sequenze ed enumerazione). Attività di laboratorio: familiarizzazione con l'ambiente di programmazione ad oggetti, elementi necessari di Java o C++, implementazione di progetti guidati di piccole/medie dimensioni.

Inglese

L'insegnamento introduce le strutture linguistiche da un punto di vista morfologico, sintattico e semantico in relazione a situazioni specifiche e contesti culturali inglesi. Avvia alla comunicazione e comprensione della lingua inglese sia scritta che orale. All'interno dell'insegnamento si faranno letture ed analisi di testi scientifici. Obiettivo dell'insegnamento è preparare all'esposizione orale e scritta, con attività diversificate quali: riassunto, rielaborazione, interpretazione e formulazione di istruzioni, in riferimento ai testi studiati.

5.2) Insegnamenti di secondo anno

Algebra

Aritmetica modulare. I numeri interi. L'algoritmo di Euclide. Congruenze ed equazioni in \mathbb{Z}_n . Il teorema di Eulero-Fermat. Calcolo combinatorio. Problemi di enumerazione. Principio della somma e del prodotto. Tecniche di conteggio. Permutazioni. Combinazioni. Combinazioni con ripetizione. Coefficienti binomiali ed il triangolo di Pascal. Il principio di inclusione-esclusione.

Si introdurranno poi elementi di teoria dei gruppi (gruppi ciclici, gruppo simmetrico e omomorfismi di gruppi), dei sistemi lineari (condizioni di compatibilità e metodi di risoluzione), degli spazi vettoriali (reali, sottospazi, dipendenza lineare, basi). Trasformazioni lineari delle matrici (matrici associate a una trasformazione lineare, diagonalizzazione).

Basi di dati

Introduzione ai sistemi di gestione di basi di dati. Cenni storici. Aspetti caratterizzanti dei sistemi di gestione di basi di dati. Evoluzione di modelli e sistemi. Il modello relazionale. Concetti di base: dominio, attributo, relazione, n-upla, schema. I linguaggi di interrogazione (algebra relazionale, linguaggi relazionalmente completi). Teoria della normalizzazione. Dipendenze funzionali. Assiomi di Armstrong e chiusura di un insieme di dipendenze. Forme normali. Scomposizioni conservative. L'organizzazione fisica dei dati. La memoria secondaria. Record fisici e record logici. Puntatori. Blocchi. File heap. File hash. File con indice (indici densi e indici sparsi). B-tree. La gestione della concorrenza, transazioni. Schedule seriale. Serializzabilità. Modelli di transazioni e meccanismi di locking. Livelock e deadlock. Protocolli a due fasi. Protocolli conservativi e aggressivi. Dati "sporchi". Rollback a cascata. Timestamp. Interrogazione di una base di dati relazionale. Il linguaggio SQL. Progettazione di una base di dati. Fasi, strumenti e metodologie di progettazione. Raccolta e analisi dei requisiti. Progettazione concettuale: il modello Entity/Relationship (ER). Progettazione logica: trasformazione di uno schema ER in uno schema relazionale. Progettazione fisica.



Calcolo delle probabilità

Richiami di teoria degli insiemi ed analisi combinatoria: cardinalità di Insiemi numerabili e non numerabili. Permutazioni, combinazioni, coefficienti binomiali, teorema del binomio, coefficienti multinomiali, teorema multinomiale, numero di soluzioni intere (non negative o positive) dell'equazione $x_1+x_2+\dots+x_r = n$, numero di modi in cui si possono selezionare r oggetti da un insieme di n oggetti distinti se vale o meno l'ordine, se c'è o meno rimpiazzamento, principio di inclusione-esclusione. Spazi di probabilità: serie a termini non negativi, spazio campionario, approccio frequentistico alla probabilità (cenni), approccio assiomatico alla probabilità. L'algebra degli eventi, funzione di probabilità, spazio di probabilità, principio di inclusione/esclusione, successioni monotone di eventi, spazi di probabilità con esiti equiprobabili, caratterizzazione della funzione di probabilità tramite i valori assunti sugli eventi elementari nel caso di spazio campionario numerabile, esempi di spazi di probabilità non numerabili. Probabilità e indipendenza: probabilità condizionata, formula di Bayes, $P(A|F)$ è una probabilità, rapporto a favore, eventi indipendenti, prove. Modellizzazione di esperimenti indipendenti: spazio di probabilità prodotto, schema di Bernoulli. Variabili aleatorie discrete: densità discreta, funzione di distribuzione, speranza matematica, momenti di ordine, varianza, deviazione standard, speranza matematica di una funzione di una v.a. discreta, formula di Stirling, v.a. uniforme, v.a. di Bernoulli, v.a. binomiale, v.a. geometrica, v.a. di Poisson, v.a. ipergeometrica, approssimazione di una v.a. binomiale tramite v.a. di Poisson, assenza di memoria della variabile geometrica.

Progettazione di algoritmi

Strutture dati avanzate (grafi e loro varianti): algoritmi fondamentali ed analisi della loro efficienza. Tecniche di progettazione di algoritmi: approccio greedy (problemi, algoritmi emblematici, correttezza e analisi). Tecniche di progettazione di algoritmi: approccio divide et impera e programmazione dinamica (problemi, algoritmi emblematici, correttezza ed analisi). Tecniche di progettazione di algoritmi: backtracking e metodi branch and bound (problemi, algoritmi emblematici, correttezza ed analisi). Nozioni di base sulle classi di complessità.

Reti di elaboratori

L'insegnamento illustra le problematiche fondamentali, le relative soluzioni ed il funzionamento delle moderne reti di elaboratori. In particolare vengono presentate le caratteristiche principali dei livelli fisico, data link, rete, trasporto ed applicazione. Per ciascun livello, inoltre, vengono descritte le caratteristiche ed è illustrato il funzionamento dei principali protocolli impiegati nella rete Internet. Infine, vengono presentate le tecniche principali, con particolare riferimento ai linguaggi di programmazione orientati ad oggetti, per lo sviluppo di applicazioni di rete di tipo client-server.

Sistemi operativi

L'insegnamento introduce i principi, l'architettura, e il progetto di un sistema operativo, segue un approccio descrittivo a livello introduttivo e affronta gli argomenti di programmazione concorrente, gestione della memoria, gestione del processore, gestione dei dispositivi e gestione del file system. L'obiettivo è la comprensione dei meccanismi di coordinamento dell'accesso alle risorse del calcolatore da parte dei programmi. Inoltre fornisce le nozioni necessarie per utilizzare un sistema Linux attraverso i principali comandi di base; descrivere le modalità di interazione con la shell e le sue funzionalità; insegnare le basi per la programmazione in linguaggio script (bash); introdurre la programmazione avanzata in ambiente Linux mediante system call; insegnare a scrivere programmi che utilizzino correttamente le system call per l'allocazione di memoria, l'accesso ai file e il controllo dei processi.



5.3) Insegnamenti di terzo anno

Ingegneria del software

Obiettivo dell'insegnamento è far apprendere agli studenti le metodologie e gli strumenti per la gestione dei processi software, con particolare riferimento alle metriche software ed all'analisi dei rischi. Particolare attenzione viene dedicata alle metodologie di analisi e progettazione orientate agli oggetti, e alla loro gestione e documentazione mediante il linguaggio UML. Cenni sulle tecniche di testing di programmi. E' prevista la produzione, in gruppo, di un documento di progetto per un sistema informativo di medie dimensioni.

5.3.1) Insegnamenti metodologici

Combinatoria per Informatica

La combinatoria si occupa dell'analisi descrittiva e dell'enumerazione di famiglie finite di insiemi finiti senza ulteriore struttura interna. In particolare: grafi. Alberi: teorema di Cayley sul numero degli alberi di copertura. Il codice di Prüfer. Decomposizione ciclica di permutazioni, parità di permutazioni. Colorazione: grafi bipartiti. Numero cromatico. Limitazioni per il numero cromatico e il numero di stabilità in termini del grado massimo. L'esistenza di grafi con "piccolo" numero di cliques ma di "grande" numero cromatico. Ipergrafi: teoremi di Sperner e di Erdős–Ko–Rado. Tecniche di conteggio: il principio di inclusione–esclusione. Il numero delle permutazioni "sconvolgenti" (derangement).

Fisica

L'insegnamento ha l'obiettivo di illustrare, con esempi tratti dalla fisica, il metodo di indagine scientifico. Vengono introdotti i principi e i metodi della fisica classica attraverso l'illustrazione delle leggi fondamentali della meccanica classica e dell'elettromagnetismo classico. Gli studenti imparano a impostare e risolvere, analiticamente e numericamente, semplici problemi di meccanica ed elettromagnetismo, acquisendo così un comportamento scientifico nell'individuazione dei problemi e nell'adozione di strategie volte alla loro analisi e alla loro soluzione. Inoltre, gli studenti potranno verificare come l'uso di strumenti informatici sia indispensabile nella soluzione di problemi di semplice impostazione, ma la cui soluzione analitica è molto difficile, se non impossibile.

Intelligenza artificiale

Introduzione. Breve descrizione delle branche dell'IA. Aree metodologiche: strategie di ricerca, risoluzione di problemi, rappresentazione della conoscenza, ragionamento automatico, apprendimento automatico. Aree applicative: elaborazione del linguaggio naturale, robotica, visione artificiale, sistemi esperti. Metodi di ricerca: strategie di ricerca nello spazio degli stati, DFS, BFS, best-first search, A*, euristiche. Rappresentazione della conoscenza: richiami di logica del prim'ordine, reti semantiche, grafi concettuali, rappresentazioni strutturate della conoscenza (frame, script), ontologie. Ragionamento automatico: richiami di logica del prim'ordine, risoluzione, frammenti della logica del prim'ordine: logiche descrittive. Strumenti per il ragionamento automatico: racer. Apprendimento automatico: apprendimento da esempi, apprendimento per rinforzo. Progetto di un sistema esperto.

Modelli e ottimizzazione

Problemi e modelli: problemi decisionali, esistenziali e di ottimizzazione. Modelli analitici di problemi reali e loro codifica. Variabili logiche, intere, reali. Vincoli. Esempi: pianificazione della produzione, il problema dello zaino, MST, TSP, SAT, ordinamento dei lavori su macchine e minimizzazione del numero delle macchine. Problemi di partizioni e problemi di riempimento. Progetto di una rete. Vincoli di soglia. Programmazione lineare: geometria della



programmazione lineare. Dualità. Teorema della dualità (debole e forte). Lemma di Farkas. Algoritmo del simplesso. Esempi. Reti di flusso: Problemi di flusso su grafi. Algoritmi e loro applicazioni a problemi concreti.

Paradigmi di programmazione

Evoluzione dei linguaggi di programmazione: dai linguaggi assembler ai linguaggi ad alto livello. Macchine astratte, interpreti, compilatori. Sintassi (BNF) e semantica (SOS: cenni). Ambiente, regole di scoping e loro implementazione. Pila dei record di attivazione; heap. Gestione della memoria: garbage collection. Controllo di sequenza, procedure, ricorsione. Tipi e controllo di tipo. Parametri e modalità di passaggio: per valore, per riferimento, per risultato, per nome. Parametri funzionali. Chiusure. Eccezioni. Il paradigma object-oriented: classi e oggetti, inizializzazione, inheritance e late-binding, il modello operativo, differenza tra subtyping e inheritance. Il paradigma funzionale: la semantica operativa, valutazione lazy e call-by-value, i tipi, i tipi polimorfi, subtyping, algoritmi di inferenza di tipi, i tipi principali. Il paradigma logico: unificazione, derivazioni SLD, semantica operativa, interpretazione dichiarativa, cenni ad aspetti implementativi.

Sicurezza

Controllo degli accessi: modelli e risultati fondazionali. Politiche di sicurezza: confidenzialità. Integrità. Politiche ibride. Reference monitor. Tecniche crittografiche: introduzione alla crittografia. Funzioni hash e message digest. Crittografia simmetrica. Crittografia asimmetrica. Protocolli crittografici. Tecniche di autenticazione: fondamenti. Punti di forza e vulnerabilità. Password. Challenge response. Biometria. Composizione di metodi. Tecnologie per la sicurezza delle reti: firewall. Sicurezza dei sistemi di e-mail. SSL. SET. Intrusion detection systems.

5.3.2) Insegnamenti tecnologici

Interazione uomo macchina

L'insegnamento affronta i problemi dell'uso di programmi da parte di persone che non hanno una cultura tecnica. In passato si richiedeva una minima conoscenza della tecnologia e manuali adeguati per l'uso di questi programmi, ora si tende a progettare sistemi e programmi tenendo conto delle capacità umane (di apprendimento, di memorizzazione, di riconoscimento e di percezione) per sfruttarle nel miglior modo possibile.

Programmazione di sistema

L'insegnamento ha come principale scopo presentare le più importanti tecniche di programmazione di sistema in ambiente Linux e Windows. Vengono confrontate le scelte progettuali e di implementazione delle chiamate di sistema per sottolineare le somiglianze ma anche le differenze tra i due ambienti che risultano importanti in fase di sviluppo di applicazioni reali.

Programmazione per il Web

L'insegnamento affronta le principali metodologie di programmazione per il Web, dalla programmazione lato client a quella lato server. Vengono illustrati nel dettaglio i linguaggi JavaScript e JSP e viene introdotta la programmazione di servlet. Le lezioni sono sempre accompagnate da esercitazioni pratiche durante le quali gli studenti svolgono un progetto che li impegna durante tutto l'insegnamento.



Sistemi di basi di dati

Questo insegnamento è fortemente integrato con quello di Basi di dati: si approfondiscono i sistemi di gestione di basi di dati, l'ottimizzazione delle interrogazioni, i sistemi distribuiti (ottimizzazione delle interrogazioni, gestione della concorrenza), i modelli e linguaggi basati sulla logica e orientati agli oggetti.

5.3.3) Insegnamenti complementari

Calcolo numerico

Si acquisiranno gli strumenti matematici di base per risolvere in modo approssimato problemi complessi di interesse nelle applicazioni. In particolare si analizzeranno metodi numerici per l'algebra lineare, metodi di approssimazione, risoluzione numerica di equazioni differenziali, algoritmi per la trasformata di Fourier e wavelet.

Economia aziendale

Si inquadra lo sviluppo storico e lo scenario attuale dell'economia mondiale, europea ed italiana, si illustrano le forme societarie delle imprese, i modelli organizzativi e le tendenze in atto. Vengono presentati i processi per la pianificazione strategica e la gestione aziendale, i modelli per la pianificazione e la gestione operativa delle aree aziendali e per la definizione dei requisiti dei sistemi informativi. Si introducono le tecniche principali per la reingegnerizzazione dei processi.

Informatica e diritto

L'insegnamento mira a formare professionalità informatiche orientate ai profili di interesse giuridico relativamente all'analisi delle fondamentali problematiche e implicazioni giuridiche dell'informatica. Esso è costituito da un nucleo centrale di lezioni di base nelle quali sono sviluppati gli argomenti essenziali e comuni, alle quali si affiancano due moduli seminariali ed integrativi di approfondimento, dedicati rispettivamente ai temi civili e penali nel rapporto tra informatica e diritto.