

Elenco corsi, a.a. 2005/2006

Corso di Laurea in Informatica

## Algebra A

Codice: AA005 Crediti: 6 Semestre: 2 Sigla: AL

Settore disciplinare: MAT/02 - Algebra

### Docente

Francesco Paolo Di Stefano  
*distefan@dm.unipi.it*  
Tel. 050-2213238

### Obiettivi di apprendimento

Il corso si propone di presentare alcune strutture algebriche tramite gli esempi degli interi, dei polinomi, delle matrici, approfondendo in particolare alcune tematiche di algebra lineare.

### Descrizione

Si presentano le strutture di gruppo, anello e campo con esempi tratti dall'aritmetica degli interi e dei polinomi. Il corso si concentra anche su alcune questioni di algebra lineare, quali risoluzione di sistemi lineari, spazi vettoriali e applicazioni lineari, diagonalizzazione di matrici.

### Programma

1. Strutture algebriche. Nozione di gruppo, anello e campo; esempi. L'anello degli interi. I polinomi: teorema di divisione, radici, teorema di Ruffini.
2. Sistemi e matrici. Operazioni sulle matrici, operazioni elementari per riga e per colonna. Sistemi lineari, algoritmo di Gauss. Matrici invertibili e calcolo della matrice inversa.
3. Algebra lineare. Spazi vettoriali, sottospazi. Basi e coordinate. Dimensione di uno spazio vettoriale. Rango. Applicazioni lineari e matrici associate, nucleo e immagine e formula delle dimensioni. Cambiamenti di basi. Dimensione dello spazio delle soluzioni di un sistema lineare omogeneo.

### Bibliografia

- M. Abate, "Algebra lineare", McGraw-Hill.
- L. Childs, "Algebra", ETS editrice.

## Modalità di esame

Scritto e orale

---

## Algebra B

Codice: AA005 Crediti: 6 Semestre: 2 Sigla: AL

Settore disciplinare: MAT/02 - Algebra

### Docente

Pier Daniele Napolitani  
*napolita@dm.unipi.it*  
Tel. 050/2213273

### Obiettivi di apprendimento

Il corso si propone di presentare alcune strutture algebriche tramite gli esempi degli interi, dei polinomi, delle matrici, approfondendo in particolare alcune tematiche di algebra lineare.

### Programma

1. Strutture algebriche. Nozione di gruppo, anello e campo; esempi. L'anello  $\mathbb{Z}_n$ . I polinomi: teorema di divisione, radici, teorema di Ruffini.
2. Sistemi e matrici. Operazioni sulle matrici, operazioni elementari per riga e per colonna. Sistemilineari, algoritmo di Gauss. Matrici invertibili e calcolo della matrice inversa.
3. Algebra lineare. Spazi vettoriali, sottospazi. Basi e coordinate. Dimensione di uno spazio vettoriale. Rango, teorema di Rouchè-Capelli. Applicazioni lineari e matrici associate, nucleo e immagine e formula delle dimensioni. Cambiamenti di basi. Dimensione dello spazio delle soluzioni di un sistema lineare omogeneo. Determinante e sue proprietà. Applicazioni e matrici diagonalizzabili, autovalori e autovettori, teorema di diagonalizzazione.

### Bibliografia

M. Abate, "Algebra lineare", McGraw-Hill

### Modalità di esame

Scritto e orale

---

## Algebra C

Codice: AA005 Crediti: 6 Semestre: 2 Sigla: AL

Settore disciplinare: MAT/02 - Algebra

### Docente

Giovanni Gaiffi  
*gaiffi@dm.unipi.it*  
*www.dm.unipi.it/~gaiffi/*  
*Tel. 050-2213242*

### Prerequisiti

Si consiglia di conoscere gli argomenti trattati nel corso di Linguaggio e Metodi della Matematica.

### Obiettivi di apprendimento

Il corso si propone di presentare alcune strutture algebriche tramite gli esempi degli interi, dei polinomi, delle matrici, approfondendo in particolare alcune tematiche di algebra lineare.

### Programma

1. Strutture algebriche. Nozione di gruppo, anello e campo; esempi. L'anello  $\mathbb{Z}_n$ . I polinomi: teorema di divisione, radici, teorema di Ruffini. Costruzione di nuovi anelli e campi come quozienti di anelli di polinomi.
2. Sistemi e matrici. Operazioni sulle matrici, operazioni elementari per riga e per colonna. Sistemi lineari, algoritmo di Gauss. Matrici invertibili e calcolo della matrice inversa.
3. Algebra lineare. Spazi vettoriali, sottospazi. Basi e coordinate. Dimensione di uno spazio vettoriale. Rango, teorema di Rouchè-Capelli. Applicazioni lineari e matrici associate, nucleo e immagine e formula delle dimensioni. Cambiamenti di basi. Dimensione dello spazio delle soluzioni di un sistema lineare omogeneo. Determinante e sue proprietà. Applicazioni e matrici diagonalizzabili, autovalori e autovettori, teorema di diagonalizzazione.

Ore lezione: 20 Ore esercitazione: 20

### Bibliografia

Appunti e dispense del corso. Testi consigliati: Marco Abate, "Algebra Lineare", McGraw-Hill. Lindsay Childs, "Algebra", ETS editrice.

### Modalità di esame

Scritto e orale

Ulteriore pagina web del corso: [http://www.dm.unipi.it/~gaiffi/algebra\\_2006/](http://www.dm.unipi.it/~gaiffi/algebra_2006/)

---

# Algoritmi per internet e web: crittografia

Codice: AA489 Crediti: 6 Semestre: 1 Sigla: CRI

Settore disciplinare: INF/01 - Informatica

## Docente

Fabrizio Luccio  
*luccio@di.unipi.it*  
*www.di.unipi.it/~luccio*  
*stanza n.278, dip.to di Informatica*  
*tel: 0502212720*

## Prerequisiti

Nozioni di base di algebra, teoria della probabilità, algoritmi e strutture dati, sistemi operativi. Gli studenti dei primi due anni della laurea triennale in informatica sono sconsigliati a frequentare il corso.

## Obiettivi di apprendimento

Introdurre le nozioni elementari sottostanti il progetto dei sistemi di cifratura moderni.

## Descrizione

Definire i concetti di Crittografia e Crittoanalisi. Definire la nozione di cifrario perfetto e discutere gli aspetti legati ad un suo possibile progetto (p.e. one-time pad). Introdurre i concetti di cifrario a chiave segreta (o simmetrico, p.e. DES e AES) e cifrario a chiave pubblica (o asimmetrico, p.e. RSA). Capire le loro proprietà e i loro limiti. Introdurre i concetti di firma digitale e autenticazione elettronica, e le loro possibili realizzazioni alla luce dei metodi precedenti. Introdurre concetti più avanzati relativi alla steganografia, i virus, le possibilità offerte dalla crittografia quantistica.

## Programma

- Introduzione: definizione di crittografia e crittoanalisi.
- Cifrari Storici
- Generatori di numeri pseudo-casuali
- Cifrari perfetti: definizione e proprietà, il One-time pad.
- Cifrari a chiave simmetrica: DES, Triple-DES e AES
- Cifrari composti
- Cifrari a chiave pubblica: funzioni one-way trapdoor e RSA.
- Identificazione, Autenticazione e Firma digitale.
- Il sistema SSL.
- Smartcard: costituzione e applicazioni.
- Elementi di steganografia.
- Virus e simili attacchi.

- Elementi di crittografia quantistica.

## Bibliografia

1. Paolo Ferragina e Fabrizio Luccio. Crittografia: principi, algoritmi e applicazioni, Bollati Boringhieri, Luglio 2001.
2. Articoli e dispense fornite dal docente durante il corso.

## Modalità di esame

Scritto e eventuale orale per spiegare lo scritto

---

## Algoritmica A

Codice: AA006   Crediti: 9   Semestre: 1   Sigla: ALG

Settore disciplinare:            INF/01 - Informatica

### Docente

Paolo Ferragina  
*ferragin@di.unipi.it*  
*www.di.unipi.it/~ferragin*  
*stanza n.359, dip.to di Informatica*  
*tel:0502212764*

### Prerequisiti

L'aver seguito i corsi di "Laboratorio di Introduzione alla Programmazione", "Fondamenti di programmazione", "Metodologie di Programmazione" e "Linguaggi e metodi della matematica".

### Obiettivi di apprendimento

Definire formalmente la nozione di algoritmo e di modello di calcolo. Caratterizzare i dati da elaborare, organizzandoli e strutturandoli nel modo più opportuno al fine di agevolarne l'uso da parte degli algoritmi. Progettare algoritmi corretti (che risolvono cioè sempre e solo il problema a cui si è interessati) ed efficienti (cioè che lo risolvono il più velocemente possibile o usano il minor spazio di memoria possibile), attraverso l'esame di diversi paradigmi. Studiare le limitazioni inerenti dei problemi da risolvere, in particolare di quelli la cui soluzione richiede l'esame di tutte le possibilità.

*Conoscenze.* Cosa è un algoritmo e cosa è un modello di calcolo. Saper caratterizzare i dati da elaborare strutturandoli nel modo più opportuno al fine da agevolarne l'uso da parte degli algoritmi. Conoscere le strutture dati e le tecniche fondamentali per il progetto di algoritmi elementari. Conoscere le tecniche di valutazione della complessità di un algoritmo e le limitazioni inerenti

dei problemi da risolvere. Conoscere le classi di complessità P, NP, NPC, e le limitazioni del calcolo (indecidibilità).

*Capacità.* Saper progettare algoritmi corretti (che risolvono cioè sempre e solo il problema a cui si è interessati) ed efficienti (cioè che lo risolvono il più velocemente possibile o usano il minor spazio di memoria possibile), attraverso l'uso di strutture dati e tecniche algoritmiche elementari. Saper valutare la complessità di un algoritmo in tempo e spazio, al caso pessimo e al caso medio. Saper valutare le limitazioni inerenti del calcolo.

*Comportamenti.* Lo studente saprà essere in grado di valutare la bontà (leggi efficienza in tempo e spazio) di un software sulla base delle sue caratteristiche progettuali salienti, prescindendo dunque dalla macchina, dal Linguaggio di sviluppo o dal Sistema Operativo sul quale il software gira. Allo stesso modo, lo studente saprà essere in grado di effettuare scelte progettuali ragionevoli nella realizzazione di un software, valutando le loro implicazioni sulla prestazioni dello stesso, prima di arrivare alla fase di programmazione, debugging e profiling, mediante l'uso di modelli matematici opportuni. Ciò risulta cruciale per ridurre i tempi e i costi dello sviluppo del software.

## Descrizione

Introduzione ai modelli di calcolo, all'analisi e alla complessità degli algoritmi. Algoritmi ricorsivi e relazioni di ricorrenza. Algoritmi di ordinamento. Strutture dati elementari e avanzate. Algoritmi su grafi e stringhe. Enumerazione e non determinismo. Problemi P, NP, RP, e NP-completi.

## Indicazioni metodologiche

Per conseguire gli obiettivi indicati, sarà necessario:

- organizzare il processo di apprendimento in moduli flessibili, posti in sequenza logica, e di difficoltà crescente;
- partire dal problem solving per arrivare alla proposta di soluzioni algoritmiche efficienti;
- presentare in modo approfondito ogni specifica struttura dati o tecnica algoritmica, confrontandola con quelle elementari viste nei corsi di Laboratorio, così da apprezzare le loro potenzialità;
- partire dalla descrizione di problemi reali per arrivare allo studio della loro complessità inerente, dimostrando così i limiti del calcolo;

## Programma

- Analisi di Algoritmi e Complessità
  - Introduzione alla complessità di calcolo
  - Ordini di grandezza delle funzioni
  - Algoritmi ricorsivi e Tecnica divide&conquer
  - Esempi su: Numeri di Fibonacci, Calcolo del massimo e minimo, Ricerca Binaria, Mergesort, Moltiplicazione rapida di interi.
  - Relazioni di ricorrenza e Teorema Principale
  - Limiti inferiori alla complessità
- Progetto di Algoritmi e Strutture Dati
  - Code, Pile, Liste e Alberi

- Heap e Heapsort
- Quicksort deterministico e randomizzato
- Ordinamento in tempo lineare: Counting sort, Radix sort, Bucket sort
- Tabelle hash: Liste di concatenazione, Indirizzamento aperto, Hashing Perfetto
- Alberi binari di ricerca
- Alberi delta-bilanciati e AVL: definizione e operazioni di rotazione
- Grafi: rappresentazione, visita DFS e BFS, topological sort
- Classi di Complessità
  - Enumerazione e non determinismo: combinazioni e permutazioni
  - Classi P, NP, NPC, NP-hard
- Modelli di Calcolo e Calcolabilità
  - Indecidibilità e universalità
  - Macchina di Turing

Ore lezione: 36 Ore esercitazione: 36

### Bibliografia

Il corso farà riferimento a due libri di testo, più una serie di appunti disponibili sul sito del corso:

1. T. Cormen, C. Leiserson, R. Rivest, "Introduzione agli algoritmi e strutture dati", McGraw-Hill Italia, 2005.
2. F. Luccio, La Struttura degli Algoritmi, Boringhieri, 1984.

### Modalità di esame

L'esame di Algoritmica consiste di una prova scritta e di una prova orale. Durante la prova scritta gli studenti non possono consultare i propri libri e appunti. L'esame scritto dunque consiste di esercizi e domande teoriche. Durante il corso vengono svolte due verifiche intermedie (compitini). Gli studenti che superano entrambe le verifiche, con una votazione superiore al 18, possono sostenere la prova orale in uno dei due appelli della sessione invernale (Gennaio o Febbraio). Chi avesse riportato la sufficienza in un compitino soltanto potrà recuperare la parte insufficiente in uno degli appelli della sessione invernale. In questi appelli la prova scritta sarà composta da due parti così da consentire il recupero suddetto. Il recupero può essere anche tentato da chi desidera migliorare il voto riportato in un compitino.

Ulteriore pagina web del corso:

<http://roquefort.di.unipi.it/~ferrax/Teach/Algoritmica.html>

## Algoritmica B

Codice: AA006 Crediti: 9 Semestre: 1 Sigla: ALG

Settore disciplinare: INF/01 - Informatica

## Docente

Roberto Grossi  
*grossi@di.unipi.it*  
*www.di.unipi.it/~grossi*  
*stanza n.361, dip.to di Informatica*  
*tel:0502212793*

## Prerequisiti

seguito i corsi di "Laboratorio di Introduzione alla Programmazione", "Fondamenti di programmazione", "Metodologie di Programmazione" e "Linguaggi e metodi della matematica".

## Obiettivi di apprendimento

Definire formalmente la nozione di algoritmo e di modello di calcolo. Caratterizzare i dati da elaborare, organizzandoli e strutturandoli nel modo più opportuno al fine di agevolarne l'uso da parte degli algoritmi. Progettare algoritmi corretti (che risolvono cioè sempre e solo il problema a cui si è interessati) ed efficienti (cioè che lo risolvono il più velocemente possibile o usano il minor spazio di memoria possibile), attraverso l'esame di diversi paradigmi. Studiare le limitazioni inerenti dei problemi da risolvere, in particolare di quelli la cui soluzione richiede l'esame di tutte le possibilità.

*Conoscenze.* Cosa è un algoritmo e cosa è un modello di calcolo. Saper caratterizzare i dati da elaborare strutturandoli nel modo più opportuno al fine da agevolarne l'uso da parte degli algoritmi. Conoscere le strutture dati e le tecniche fondamentali per il progetto di algoritmi elementari. Conoscere le tecniche di valutazione della complessità di un algoritmo e le limitazioni inerenti dei problemi da risolvere. Conoscere le classi di complessità P, NP, NPC, e le limitazioni del calcolo (indecidibilità).

*Capacità.* Saper progettare algoritmi corretti (che risolvono cioè sempre e solo il problema a cui si è interessati) ed efficienti (cioè che lo risolvono il più velocemente possibile o usano il minor spazio di memoria possibile), attraverso l'uso di strutture dati e tecniche algoritmiche elementari. Saper valutare la complessità di un algoritmo in tempo e spazio, al caso pessimo e al caso medio. Saper valutare le limitazioni inerenti del calcolo.

*Comportamenti.* Lo studente saprà essere in grado di valutare la bontà (leggi efficienza in tempo e spazio) di un software sulla base delle sue caratteristiche progettuali salienti, prescindendo dunque dalla macchina, dal Linguaggio di sviluppo o dal Sistema Operativo sul quale il software gira. Allo stesso modo, lo studente saprà essere in grado di effettuare scelte progettuali ragionevoli nella realizzazione di un software, valutando le loro implicazioni sulla prestazioni dello stesso, prima di arrivare alla fase di programmazione, debugging e profiling, mediante l'uso di modelli matematici opportuni. Ciò risulta cruciale per ridurre i tempi e i costi dello sviluppo del software.

## Descrizione

Introduzione ai modelli di calcolo, all'analisi e alla complessità degli algoritmi.

Algoritmi ricorsivi e relazioni di ricorrenza. Algoritmi di ordinamento. Strutture dati elementari e avanzate. Algoritmi su grafi e stringhe. Enumerazione e non determinismo. Problemi P, NP, RP, e NP-completi.

### Indicazioni metodologiche

Per conseguire gli obiettivi indicati, sarà necessario:

- organizzare il processo di apprendimento in moduli flessibili, posti in sequenza logica, e di difficoltà crescente;
- partire dal problem solving per arrivare alla proposta di soluzioni algoritmiche efficienti;
- presentare in modo approfondito ogni specifica struttura dati o tecnica algoritmica, confrontandola con quelle elementari viste nei corsi di Laboratorio, così da apprezzare le loro potenzialità;
- partire dalla descrizione di problemi reali per arrivare allo studio della loro complessità inerente, dimostrando così i limiti del calcolo;

### Programma

- Analisi di Algoritmi e Complessità
  - Introduzione alla complessità di calcolo
  - Ordini di grandezza delle funzioni
  - Algoritmi ricorsivi e Tecnica divide&conquer
  - Esempi su: Numeri di Fibonacci, Calcolo del massimo e minimo, Ricerca Binaria, Mergesort, Moltiplicazione rapida di interi.
  - Relazioni di ricorrenza e Teorema Principale
  - Limiti inferiori alla complessità
- Progetto di Algoritmi e Strutture Dati
  - Code, Pile, Liste e Alberi
  - Heap e Heapsort
  - Quicksort deterministico e randomizzato
  - Ordinamento in tempo lineare: Counting sort, Radix sort, Bucket sort
  - Tabelle hash: Liste di concatenazione, Indirizzamento aperto, Hashing Perfetto
  - Alberi binari di ricerca
  - Alberi delta-bilanciati e AVL: definizione e operazioni di rotazione
  - Grafi: rappresentazione, visita DFS e BFS, topological sort
- Classi di Complessità
  - Enumerazione e non determinismo: combinazioni e permutazioni
  - Classi P, NP, NPC, NP-hard
- Modelli di Calcolo e Calcolabilità
  - Indecidibilità e universalità
  - Macchina di Turing

Ore lezione: 36 Ore esercitazione: 36

### Bibliografia

Il corso farà riferimento a due libri di testo, più una serie di appunti disponibili sul sito del corso:

1. T. Cormen, C. Leiserson, R. Rivest, "Introduzione agli algoritmi e strutture dati", McGraw-Hill Italia, 2005.
2. F. Luccio, La Struttura degli Algoritmi, Boringhieri, 1984.

### Modalità di esame

L'esame di Algoritmica consiste di una prova scritta e di una prova orale. Durante la prova scritta gli studenti non possono consultare i propri libri e appunti. L'esame scritto dunque consiste di esercizi e domande teoriche. Durante il corso vengono svolte due verifiche intermedie (compitini). Gli studenti che superano entrambe le verifiche, con una votazione superiore al 18, possono sostenere la prova orale in uno dei due appelli della sessione invernale (Gennaio o Febbraio). Chi avesse riportato la sufficienza in un compitino soltanto potrà recuperare la parte insufficiente in uno degli appelli della sessione invernale. In questi appelli la prova scritta sarà composta da due parti così da consentire il recupero suddetto. Il recupero può essere anche tentato da chi desidera migliorare il voto riportato in un compitino.

Ulteriore pagina web del corso: <http://www.di.unipi.it/~grossi/ALG>

---

## Analisi matematica A

Codice: AA001   Crediti: 8   Semestre: 1-2   Sigla: AM

Settore disciplinare: MAT/05 - Analisi Matematica

### Docente

Elvira Dalla Vedova  
*dallaved@dm.unipi.it*  
Tel. 050-2213262

### Prerequisiti

Nessuno

### Obiettivi di apprendimento

Fornire quel minimo di strumenti di analisi matematica che sono indispensabili per qualunque studente della facoltà di Scienze M. F. N.

### Descrizione

Dopo aver introdotto il sistema dei numeri reali, e illustrato un minimo di geometria analitica nel piano anche con l'ausilio dei numeri complessi, si descrivono le idee fondamentali del calcolo, sia dal punto di vista "discreto" (successioni e serie), sia "continuo" (calcolo differenziale, calcolo integrale, cenni sulle equazioni differenziali).

## Programma

Alfabeto greco. Insiemi. Numeri reali e loro proprietà. Numeri naturali, interi, razionali e loro proprietà. Radici n-sime, esponenziale, logaritmi. Numeri complessi. Geometria analitica nel piano. Funzioni trigonometriche. Successioni e loro limiti. Serie. Criteri di convergenza. Successioni di Cauchy, massimo e minimo limite. Serie di potenze. Moltiplicazione di serie. Funzioni reali. Limiti, continuità e proprietà collegate. Derivazione e proprietà collegate. Principio di identità delle serie di potenze. Confronto di infinitesimi e di infiniti. Formula di Taylor. Massimi e minimi relativi. Convessità. Asintoti. L'integrale e le sue proprietà. Uniforme continuità. Teorema fondamentale del calcolo integrale. Metodi di integrazione. Integrali impropri. Equazioni differenziali del primo ordine in forma normale, equazioni lineari ed a variabili separabili. Equazioni lineari del secondo ordine a coefficienti costanti.

## Bibliografia

P. Marcellini, C. Sbordone, *Analisi matematica uno*, Liguori Editore, Napoli. M. Giaquinta, G. Modica, *Analisi matematica*, vol.1 (funzioni di una variabile) e vol.2 (approssimazione e processi discreti), Pitagora Editrice, Bologna. E. Giusti, *Analisi matematica 1*, Boringhieri, Torino. P. Marcellini, C. Sbordone, *Esercitazioni di matematica*, vol.1, parte prima e parte seconda, Liguori Editore, Napoli.

## Modalità di esame

Scritto e orale

---

## Analisi matematica B

Codice: AA001   Crediti: 8   Semestre: 1-2   Sigla: AM

Settore disciplinare: MAT/05 - Analisi Matematica

### Docente

Antonio Angelo Tarsia  
*tarsia@dm.unipi.it*  
*www.dm.unipi.it/pages/tarsia/public\_html/*  
*Tel. 050-2213212*

### Prerequisiti

Polinomi. Equazioni e disequazioni del primo e del secondo ordine. Funzioni trigonometriche elementari. Proprietà elementari delle potenze e dei logaritmi.

### Obiettivi di apprendimento

Fornire quel minimo di strumenti di analisi matematica che sono indispensabili per qualunque studente della facoltà di Scienze M. F. N.

### **Descrizione**

Preliminarmente vengono studiate le funzioni elementari. Quindi viene introdotto il sistema dei numeri reali. Si descrivono poi le idee fondamentali del calcolo, sia dal punto di vista "discreto" (successioni e serie), sia "continuo" (calcolo differenziale, calcolo integrale in una variabile, equazioni differenziali lineari).

### **Indicazioni metodologiche**

Il corso si svolge in due semestri. Nel primo semestre sono previste 3 ore di attività frontale (lezioni ed esercitazioni) , 1 ora di ricevimento studenti in aula, 2 ore di corso (facoltativo) sulla matematica elementare. Nel secondo semestre sono previste 3 ore di attività frontale (lezioni ed esercitazioni) , 1 ora di ricevimento studenti in aula. Per tutto il corso dell'anno il docente è a disposizione almeno due ore alla settimana per delucidazioni sugli argomenti svolti a lezione.

Nella pagina web del docente gli studenti possono trovare esercizi svolti, oltre che il programma svolto ed aggiornato degli argomenti trattati in ogni singola lezione.

### **Programma**

Insiemi numerici. Induzione. Funzioni elementari. Numeri reali e loro proprietà. Successioni e loro limiti. Cenni sulle serie numeriche e di potenze. Funzioni reali. Limiti, continuità e proprietà collegate. Derivazione e proprietà collegate. Confronto di infinitesimi e di infiniti. Teoremi di De l'Hopital e formula di Taylor. Massimi e minimi relativi. Studio di funzioni. L'integrale in una variabile e le sue proprietà. Teorema fondamentale del calcolo integrale. Metodi di integrazione. Integrali impropri. Equazioni differenziali del primo ordine in forma normale, equazioni lineari. Equazioni lineari del secondo ordine a coefficienti costanti.

Ore lezione: 30 Ore esercitazione: 30

### **Bibliografia**

M. Sasseti - A. Tarsia , Corso preliminare di Matematica, SEU, Pisa, 2005

M. Sasseti, Calcolo - parte prima: Calcolo Differenziale, ed. Il Borghetto, 2005.

M. Sasseti, Calcolo - parte seconda: Calcolo Integrale, ed. Il Borghetto, 2005.

### **Modalità di esame**

4 prove in itinere, prova scritta finale (per quanti non hanno superato le prove in itinere), prova orale.

Ulteriore pagina web del corso:

[http://www.dm.unipi.it/pages/tarsia/public\\_html/dida.html](http://www.dm.unipi.it/pages/tarsia/public_html/dida.html)

---

## Analisi matematica C

Codice: AA001   Crediti: 8   Semestre: 1-2   Sigla: AM

Settore disciplinare:            MAT/05 - Analisi Matematica

### Docente

Mauro Sassetti  
*sassetti@dm.unipi.it*  
*www.dm.unipi.it/~sassetti/*  
*tel: 0502212700*

### Prerequisiti

La matematica di base , in particolare : equazioni e disequazioni di I e II grado , valore assoluto, potenze , radici , funzioni esponenziali , logaritmiche e trigonometriche.

### Obiettivi di apprendimento

Fornire i primi strumenti di analisi matematica sia come tecniche di calcolo che come ragionamento logico-deduttivo.

### Descrizione

Dopo aver introdotto il sistema dei numeri reali e il concetto di funzione di una variabile reale, sono esposte le idee fondamentali del calcolo differenziale e di quello integrale . Al punto di vista continuo è affiancato quello discreto ( successioni , serie ) .

### Indicazioni metodologiche

Il corso ( diviso in due semestri ) prevede settimanalmente 3 ore di attività didattica frontale , 1 ora di ricevimento studenti in aula , almeno 2 ore di ricevimento individuale. Nel primo semestre sono previste 2 ore settimanali per lo svolgimento di un corso di matematica di base .

Il corso è strutturato in moduli posti in sequenza logica , ciascuno svolto dal punto di vista teorico e pratico . Ove possibile , il punto di partenza è dato da un problema concreto .

Le lezioni sollecitano la partecipazione attiva degli studenti .

Gli argomenti trattati a lezione sono riportati in dispense a cura del docente .

Ogni modulo didattico è affiancato da materiale didattico da scaricare dall'apposita pagina personale del docente , che si aggiunge a quello dei testi indicati . In tale pagina è anche riportato il registro delle lezioni con indicazioni

dettagliate degli argomenti trattati .

Lo svolgimento del corso è coordinato con i docenti dei corsi paralleli.

Il livello di raggiungimento degli obiettivi è valutato con verifiche intermedie.

## Programma

Numeri reali e numeri complessi. Prime definizioni legate al concetto di funzione di una variabile reale. Le principali funzioni elementari .Limiti di funzioni . Infinitesimi ed infiniti. Funzioni continue e funzioni derivabili : principali proprietà. Studio di una funzione e del suo grafico. Formula di Taylor. Ricerca dei punti di massimo e di minimo. Convessità. L'integrale e le sue proprietà ; teorema fondamentale del calcolo integrale . Metodi di integrazione. Integrali impropri . Cenni su equazioni differenziali lineari del I e II ordine e a variabili separabili. Successioni e serie numeriche .

1. Numeri reali Definizione assiomatica del corpo  $\mathbb{R}$  dei numeri reali: assiomi algebrici, di ordinamento e di continuità. Sottoinsiemi di  $\mathbb{R}$  limitati superiormente o inferiormente; massimo e minimo, estremo superiore ed inferiore. Rappresentazione dei numeri reali sulla retta cartesiana ovvero retta reale. Intervalli di numeri reali. I numeri decimali ; loro densità sulla retta reale ovvero approssimazione decimale dei numeri reali; i numeri reali come allineamenti decimali propri ovvero rappresentazione decimale dei numeri reali. Principio di induzione. Formula del binomio di Newton. Intorni di un punto nella retta reale; punti interni, esterni, di frontiera e isolati per un sottoinsieme di  $\mathbb{R}$ . La retta reale estesa. Intorni di un punto nella retta reale estesa; punti di accumulazione per un sottoinsieme di  $\mathbb{R}$  .

2. Numeri complessi Definizione algebrica del corpo  $\mathbb{C}$  dei numeri complessi come coppie ordinate di numeri reali con le operazioni di somma e di prodotto. Parte reale , parte immaginaria e coniugato di un numero complesso. Rappresentazione nel piano cartesiano. Il sottocorpo dei numeri reali. Unità immaginaria. Rappresentazioni standard , trigonometrica ed esponenziale dei numeri complessi. Interpretazione geometrica delle operazioni algebriche. Radici n-esime di un numero complesso. Radici di polinomi in campo complesso e loro molteplicità. Teorema fondamentale dell'algebra. Nei polinomi a coefficienti reali le eventuali radici non reali compaiono a coppie coniugate. Scomposizione di un polinomio in campo reale e in campo complesso. Funzioni complesse di una variabile reale; l'esponenziale complesso.

3. Funzioni reali di una variabile reale Definizione di funzione reale di una variabile reale ; dominio, campo di esistenza, immagine, grafico. Primi esempi di funzioni: costanti, lineari, esprimenti proporzionalità inversa, valore assoluto, segno, polinomi, razionali, trigonometriche , esponenziali. Operazioni algebriche tra funzioni , in particolare la composizione di funzioni . Funzioni monotone e funzioni iniettive. Invertibilità di una funzione iniettiva; la funzione inversa; legame tra il grafico di una funzione invertibile e quello della sua inversa. Le funzioni logaritmiche , le potenze con esponente reale , le funzioni trigonometriche inverse. Le funzioni elementari. Funzioni limitate; estremo superiore ed inferiore, massimo e minimo di una funzione. Punti di massimo e di minimo assoluti o locali. Funzioni pari, dispari, periodiche. Restrizioni e prolungamenti. Le successioni di numeri reali come caso particolare del concetto di funzione ; definizioni collegate.

4. Limiti di una funzione Definizione generale di limite per una funzione. Limiti reali e funzioni continue in un punto. Limiti infiniti e asintoti verticali. Limiti reali

all'infinito e asintoti orizzontali. Limiti infiniti all'infinito e asintoti obliqui. Limite destro e sinistro, per eccesso e per difetto. Principali proprietà dei limiti : unicità , permanenza del segno , passaggio al limite in una disuguaglianza , confronto , restrizioni (in particolare , limite destro e sinistro) , funzioni monotone , cambiamento di variabile , operazioni con limiti reali o infiniti. Forme indeterminate. Limiti notevoli. Confronto tra infinitesimi o infiniti simultanei , parti principali, principi di sostituzione e loro uso nello studio delle forme indeterminate. Il limite di una successione come caso particolare della definizione generale di limite. Limite di successioni monotone. Il numero di Nepero. Sottosuccessioni o successioni estratte . Teorema di Bolzano-Weierstrass 5. Funzioni continue Definizione di continuità in un punto e in un insieme. Punti di discontinuità e loro classificazione. Prime proprietà delle funzioni continue (permanenza del segno, operazioni algebriche e continuità, continuità delle funzioni elementari nel loro campo di esistenza) dedotte dalle proprietà dei limiti. Proprietà delle funzioni continue in un intervallo : teorema di Weierstrass sull'esistenza del massimo e del minimo, teorema degli zeri , teorema dei valori intermedi , monotonia delle funzioni continue invertibili, continuità della funzione inversa. Applicazione del teorema degli zeri all'approssimazione degli zeri di una funzione continua . 6. Calcolo differenziale Definizione di derivata come limite del rapporto incrementale. Retta tangente. Le funzioni derivabili sono continue. Esempi di funzioni continue non derivabili. Calcolo di derivate notevoli. Regole di derivazione. Derivate successive. Il teorema di Fermat per la ricerca dei punti di massimo e di minimo. Teoremi di Rolle, Lagrange, Cauchy. Monotonia e segno della derivata. Applicazione del calcolo differenziale a problemi di massimo o minimo , allo studio del grafico di una funzione , alla risoluzione grafica di un'equazione o di una disequazione. Il teorema dell' Hôpital. Funzioni convesse e concave; punti di flesso; legami con la derivata seconda. La formula di Taylor : applicazioni allo studio locale di una funzione , al calcolo di limiti in forma indeterminata , all'approssimazione numerica. 7. Primitive di una funzione o integrale indefinito Definizione di primitiva di una funzione assegnata. Le primitive di una funzione possono non esistere ; se esistono, sono infinite e in ogni intervallo differiscono tra loro per una costante. L'integrale indefinito come insieme delle primitive di una funzione assegnata: sua notazione. Linearità dell'integrale. Integrazione per parti. Integrazione per cambiamento di variabile (con o senza l'ipotesi di invertibilità della funzione che esprime il cambiamento). Integrazione delle funzioni razionali a partire dal metodo di scomposizione di Hermite. Cambiamenti di variabile razionalizzanti. 8. Calcolo integrale Partizioni di un intervallo chiuso e limitato. Somma integrale superiore ed inferiore per una funzione limitata in un intervallo chiuso e limitato. Integrabilità di una funzione secondo la teoria di Riemann. L'integrale . Interpretazione geometrica dell'integrale : area di un sottografico o trapezoide. Estensione alla misura dell'area di regioni del piano delimitate da due grafici. Integrabilità delle funzioni continue e delle funzioni generalmente continue. Il teorema fondamentale del calcolo integrale per funzioni integrabili dotate di primitive. Il caso particolare delle funzioni continue. Le funzioni integrali. Integrazione per parti e per cambiamento di variabile. 9. Integrali in senso improprio Estensione della definizione di integrale a intervalli non limitati e a funzioni non limitate. Misura di regioni piane non limitate. Proprietà dell'integrale in senso improprio. Assoluta integrabilità. Criteri di integrabilità a priori : criterio del confronto e criterio della parte principale. Studio delle funzioni integrali. 10. Equazioni differenziali

Equazioni lineari del primo ordine : formula risolutiva , struttura algebrica dell'integrale generale e sua deduzione a priori. Equazioni lineari del secondo ordine : risultati generali . Risoluzione nel caso particolare dei coefficienti costanti : il polinomio caratteristico, il metodo dei coefficienti indeterminati e quello della variazione della costanti arbitrarie. Equazioni a variabili separate : metodo di risoluzione ed osservazioni generali . Il problema di Cauchy ovvero le condizioni iniziali. 11. Serie numeriche Definizione di serie numeriche convergenti , divergenti o indeterminate . Esempi di serie : serie armonica e serie geometrica . Condizione necessaria per la convergenza . Serie a termini di segno costante : criterio del rapporto , della radice , del confronto e di equivalenza . Convergenza assoluta . Criterio di Leibniz per le serie a segno alterno . Appendice : teoremi richiesti nella prova orale Per i teoremi preceduti dal segno (\*) la dimostrazione è facoltativa , mentre è comunque richiesta la conoscenza corretta dell'enunciato. L'elenco dei teoremi è dato seguendo la precedente divisione del programma in capitoli. 1. Le classi contigue hanno uno ed un solo elemento di separazione. Se l'estremo superiore di un insieme appartiene a tale insieme, ne è anche il massimo. I numeri razionali hanno una rappresentazione decimale finita o periodica. 2. Radici n-esime di un numero complesso. Scomposizione di un polinomio reale in fattori primi in campo complesso e in campo reale. 3. Nessuna dimostrazione, il capitolo essendo costituito quasi esclusivamente da definizioni. 4. Unicità del limite. Permanenza del segno. (\*) Limite di una funzione e delle sue restrizioni. Confronto . Passaggio al limite in una disuguaglianza. (\*) Funzioni monotone. (\*) Cambiamento di variabile . Comportamento del limite rispetto alle operazioni algebriche. (\*) Limiti notevoli. Principio di sostituzione degli infinitesimi e degli infiniti nello studio di una forma indeterminata. Relazione tra l'esistenza del limite di una successione e quello delle sue sottosuccessioni. (\*) Teorema di Bolzano-Weierstrass. Esistenza del limite per le successioni monotone. (\*) Il numero di Nepero come limite di una successione crescente e limitata. 5. Teoremi di Weierstrass , degli zeri , dei valori intermedi. Legame tra monotonia e iniettività per una funzione continua in un intervallo. Comportamento della continuità rispetto alle operazioni algebriche elementari. 6. Le funzioni derivabili sono continue. Comportamento della derivata rispetto alle operazioni algebriche elementari: somma, prodotto per un numero, prodotto, rapporto, composizione, inversione. Teorema di Fermat. Teoremi di Rolle, Lagrange e Cauchy. Segno della derivata e monotonia della funzione. Segno della derivata seconda e convessità della funzione. Condizione necessaria per un punto di flesso. Teorema dell' Hôpital (forma indeterminata  $0 / 0$  e (\*) forma indeterminata infinito / infinito ). Applicazione del teorema di Lagrange o dell'Hôpital allo studio dell'esistenza della derivata in un punto in cui la funzione è continua. La formula di Taylor con resto di Peano o (\*) di Lagrange. 7. In un intervallo le primitive differiscono per una costante. Integrazione per parti e per cambiamento di variabile. 8. Teorema della media integrale per funzioni integrabili e per funzioni continue. (\*) Integrabilità delle funzioni continue e di quelle generalmente continue. (\*) Il caso particolare delle funzioni con discontinuità eliminabile o di I specie. Teorema fondamentale del calcolo integrale nel caso generale e in quello continuo. 9. Criteri del confronto e di sostituzione della parte principale nello studio a priori di un integrale improprio. 10. Equazioni differenziali lineari del I ordine: integrale generale e deduzione a priori dei risultati generali. Equazioni differenziali lineari del II ordine: metodo del polinomio caratteristico, dei coefficienti indeterminati e della variazione

delle costanti arbitrarie. Equazioni dif-ferenziali a variabili separate: integrale generale in forma implicita e, nei casi in cui è possibile, passaggio alla forma esplicita. 11. Condizione necessaria per la convergenza . Serie a termini di segno costante : criterio del rapporto , della radice , del confronto e di equivalenza . La convergenza assoluta implica quella semplice . Criterio di Leibniz per le serie a segno alterno .

Ore lezione: 30 Ore esercitazione: 30

### **Bibliografia**

M. Sassetti - A. Tarsia : Precorso di Matematica , Ed. SEU , 2005 Pisa

M. Sassetti : Calcolo . Parte I e II , Ed. Il Borghetto , 2005 Pisa

### **Modalità di esame**

Prova scritta e successiva prova orale . Lo svolgimento positivo di compiti scritti parziali ( 2 nel primo semestre e 2 nel secondo ) può sostituire la prova scritta.

Ulteriore pagina web del corso: <http://www.dm.unipi.it/~sassetti>

---

## **Architettura degli elaboratori A**

Codice: AA008 Crediti: 10 Semestre: 1 Sigla: AE

Settore disciplinare: INF/01 - Informatica

### **Docente**

Marco Vanneschi  
*vannesch@di.unipi.it*  
*www.di.unipi.it/~vannesch*  
*stanza n.288, dip.to di Informatica*  
*tel:0502212738*

### **Prerequisiti**

Fondamenti di programmazione.

Algoritmi e strutture dati.

Basi matematiche e logiche dei corsi del primo anno.

### **Obiettivi di apprendimento**

L'obiettivo del corso di Architettura degli Elaboratori (AE) è di fornire le basi per lo studio e la conoscenza dei sistemi di elaborazione ai vari livelli e della

relazione che i sistemi hanno con gli strumenti di programmazione (linguaggi, compilatori, interpreti). Il corso approfondisce l'aspetto dei livelli assembler e firmware degli elaboratori (traduzione ed esecuzione dei programmi, unità di elaborazione, memorie, comunicazioni) arrivando a definire le caratteristiche dei processori esistenti e loro tendenze tecnologiche. In parti successive, lo studio dei processori è completato dall'architettura d'ingresso-uscita e dalle gerarchie di memoria e memorie cache. Una parte del corso è dedicata ad introdurre le caratteristiche fondamentali dei sistemi operativi, in particolare i concetti di processi cooperanti, allocazione di memoria e ingresso-uscita, curando l'aspetto delle interrelazioni tra livello del sistema operativo ed i livelli assembler e firmware, nonché con gli strumenti di programmazione.

#### *Conoscenze.*

Conoscenze di base sul tema dell'architettura degli elaboratori.

In parte, conoscenze caratterizzanti su tecnologie informatiche.

#### *Capacità.*

Capacità di apprendimento continuo.

Capacità di analisi e sintesi di concetti e tecniche, in modo consono ad un corso universitario.

Capacità di ragionamento scientifico, pur applicato ad attività a carattere tecnologico.

Capacità di presentazione di concetti e di elaborati in forma scritta e orale.

#### *Comportamenti.*

Comportamenti etici.

Comportamenti di organizzazione dello studio.

#### **Descrizione**

Il corso approfondisce l'aspetto dei livelli assembler e firmware degli elaboratori (traduzione ed esecuzione dei programmi, unità di elaborazione, memorie, comunicazioni). Lo studio dei processori è completato dall'architettura d'ingresso-uscita e dalle gerarchie di memoria e memorie cache. Una parte del corso è dedicata ad introdurre le caratteristiche fondamentali dei sistemi operativi, in particolare i concetti di processi cooperanti, allocazione di memoria e ingresso-uscita, curando l'aspetto delle interrelazioni tra livello del sistema operativo ed i livelli assembler e firmware, nonché con gli strumenti di compilazione e programmazione.

#### **Indicazioni metodologiche**

Apprendimento attraverso moduli sequenziali collegati.

Sviluppo della capacità di sintesi dei vari moduli.

Importanza delle esercitazioni sui singoli moduli e sul complesso dei moduli.

Verifiche intermedie.

### **Programma**

1. Fondamenti di strutturazione dei sistemi di elaborazione.
2. Il livello della macchina assembler.
3. Firmware e modello di unità di elaborazione. Reti logiche. Comunicazione e sincronizzazione.
4. Architettura dell'unità centrale e modello convenzionale di processore e ingresso-uscita.
5. Fondamenti di sistemi operativi: a processi comunicanti e loro supporto, allocazione dinamica di memoria, ingresso-uscita, interruzioni, eccezioni. Relazione tra sistema operativo e livelli assembler e firmware.
6. Gerarchie di memoria e architettura dell'unità centrale con cache. Cenni ad architetture ad alte prestazioni.

Ore lezione: 40 Ore esercitazione: 40

### **Bibliografia**

- M. Vanneschi, "Appunti di Architettura degli Elaboratori I". Servizio Editoriale Universitario, Università di Pisa.
- Materiale integrativo fornito durante il corso attraverso la pagina Web [www.di.unipi.it](http://www.di.unipi.it) del Prof. Vanneschi, sezione Teaching/Architettura degli Elaboratori.

Materiale di consultazione

- D.A. Patterson, J. Hennessy, "Computer Organization & Design – The Hardware/Software Interface". Morgan Kaufmann Publishers.

### **Modalità di esame**

Scritto e orale

Ulteriore pagina web del corso: <http://www.di.unipi.it/~vannesch/>

---

## **Architettura degli elaboratori B**

Codice: AA008 Crediti: 10 Semestre: 1 Sigla: AE

Settore disciplinare: INF/01 - Informatica

Docente

Marco Aldinucci  
*aldinuc@di.unipi.it*  
*www.di.unipi.it/~aldinuc*  
*stanza n.384, dip.to di Informatica*  
*tel:0502212728*

## **Prerequisiti**

Raccomandati i corsi del primo anno, e in particolare quelli di programmazione

## **Obiettivi di apprendimento**

L'obiettivo del corso di Architettura degli Elaboratori (AE) è di fornire le basi per lo studio e la conoscenza dei sistemi di elaborazione ai vari livelli e della relazione che i sistemi hanno con gli strumenti di programmazione (linguaggi, compilatori, interpreti).

Il corso approfondisce l'aspetto dei livelli assembler e firmware degli elaboratori (traduzione ed esecuzione dei programmi, unità di elaborazione, memorie, comunicazioni) arrivando a definire le caratteristiche dei processori esistenti e loro tendenze tecnologiche. In parti successive, lo studio dei processori è completato dall'architettura d'ingresso-uscita e dalle gerarchie di memoria e memorie cache. Una parte del corso è dedicata ad introdurre le caratteristiche fondamentali dei sistemi operativi, in particolare i concetti di processi cooperanti, allocazione di memoria e ingresso-uscita, curando l'aspetto delle interrelazioni tra livello del sistema operativo ed i livelli assembler e firmware, nonché con gli strumenti di programmazione.

*Conoscenze.* Lo studente acquisirà conoscenze di base, sia sui metodi che sulle tecnologie, per lo studio e la conoscenza dei sistemi di elaborazione

*Capacità.* Lo studente acquisirà le conoscenze necessarie alla valutazione delle prestazioni dei sistemi di elaborazione commerciali sulla base della piena comprensione delle soluzioni tecniche utilizzate

*Comportamenti.* Lo studente saprà analizzare le prestazioni aspettate da un sistema di calcolo commerciale mediante l'analisi delle soluzioni tecniche adottate, sapendo distinguere gli aspetti di marketing da quelli fondamentali nella progettazione dei sistemi di calcolo. Saprà inoltre distinguere quelle soluzioni di progettazione che garantiscono la progettazione di sistemi che possano garantire robustezza e protezione delle informazioni.

## **Programma**

1. Fondamenti di strutturazione dei sistemi di elaborazione.
2. Il livello della macchina assembler.
3. Firmware e modello di unità di elaborazione. Reti logiche. Comunicazione e sincronizzazione.
4. Architettura dell'unità centrale e modello convenzionale di processore e ingresso-uscita.
5. Fondamenti di sistemi operativi: a processi comunicanti e loro supporto, allocazione dinamica di memoria, ingresso-uscita, interruzioni, eccezioni. Relazione tra sistema operativo e livelli assembler e firmware.

6. Gerarchie di memoria e architettura dell'unità centrale con cache. Cenni ad architetture ad alte prestazioni.

Ore lezione: 50 Ore esercitazione: 30

### Bibliografia

- M. Vanneschi, "Appunti di Architettura degli Elaboratori I". Servizio Editoriale Universitario, Università di Pisa.
- Materiale integrativo fornito durante il corso attraverso la pagina Web [www.di.unipi.it](http://www.di.unipi.it) del Prof. Vanneschi, sezione Teaching/Architettura degli Elaboratori.

### Materiale di consultazione

- D.A. Patterson, J. Hennessy, "Computer Organization & Design – The Hardware/Software Interface". Morgan Kaufmann Publishers.

### Modalità di esame

Una prova orale preceduta da una prova scritta.

Ulteriore pagina web del corso: <http://www.di.unipi.it/~aldinuc/>

---

## Architetture parallele e distribuite

Codice: AA024 Crediti: 9 Semestre: 2 Sigla: ASE

Settore disciplinare: INF/01 - Informatica

### Docente

Marco Vanneschi  
[vannesch@di.unipi.it](mailto:vannesch@di.unipi.it)  
[www.di.unipi.it/~vannesch](http://www.di.unipi.it/~vannesch)  
stanza n.288, dip.to di Informatica  
tel: 0502212738

### Prerequisiti

Il corso assume l'esistenza della propedeuticità di *Architettura degli Elaboratori* per tutti gli aspetti riguardanti la strutturazione a livelli di interpretazione ed a moduli, architettura firmware, assembler, architettura di processori general-purpose, gerarchie di memoria, valutazione delle prestazioni, concetti di concorrenza, modelli di sistema operativo, relazioni tra architettura, sistema operativo, compilazione e supporto a tempo di esecuzione.

### Obiettivi di apprendimento

Il corso studia modelli, metodologie e tecniche per la progettazione di sistemi ed applicazioni

avanzate ad alte prestazioni. A questo scopo, viene data particolare enfasi ai modelli di rappresentazione in parallelo di computazioni secondo metodologie che siano il più possibile sistematiche e che si prestino a ricavare valutazioni di prestazioni ed ottimizzazioni sia in fase di predizione che di verifica. Le metodologie studiate permetteranno di acquisire questa padronanza, che potrà essere sfruttata tanto nel progetto di applicazioni su macchine parallele e/o sistemi distribuiti, quanto nella valutazione di soluzioni esistenti e nella loro eventuale ottimizzazione.

Il corso intende fornire una conoscenza sufficientemente approfondita, dal punto di vista tecnologico, delle principali architetture parallele e distribuite ad alte prestazioni: multiprocessor a memoria condivisa (SMP, NUMA), multicomputer, sistemi di tipo cluster di PC/workstation, con particolare enfasi sulle piattaforme abilitanti più diffuse, anche a livello commerciale, e previste nel breve termine.

Al tempo stesso, il corso pone le basi per la comprensione e lo studio delle evoluzioni che la tecnologia del calcolo ad alte prestazioni avrà nel medio-lungo termine (architetture a componenti, Grid, ambienti di sviluppo applicazioni).

Le metodologie di parallelizzazione e valutazione delle prestazioni saranno inizialmente esemplificate a sistemi a livello firmware, allo scopo di acquisire padronanza dei concetti in casi relativamente semplici da valutare, per poi passare alle computazioni parallele a livello di processi comunicanti eseguibili sulle architetture parallele e distribuite suddette.

L'aspetto dell'implementazione dei meccanismi di parallelizzazione sfrutterà tutti i più recenti risultati, scientifici e tecnologici, nel campo delle comunicazioni tra processi, processori di comunicazione e reti ad alta banda.

L'approfondimento, professionale e di ricerca, delle tematiche del corso è effettuato, in particolare, nei corsi dell'indirizzo Piattaforme ad Alte Prestazioni della Laurea Specialistica in Tecnologie Informatiche.

*Conoscenze.* Conoscenze di base nel campo delle piattaforme di calcolo ad alte prestazioni e metodologie di progettazione di applicazioni ad alte prestazioni. Conoscenze caratterizzanti le tecnologie informatiche, per quanto riguarda l'integrazione di metodologie e strumenti ad alte prestazioni nel contesto interoperabile delle tecnologie per lo sviluppo di applicazioni.

*Capacità.* Capacità di apprendimento e di presentazione di risultati dello studio / lavoro, innalzamento del livello culturale, dal punto di vista scientifico e tecnologico, in modo consono ad un corso di studi universitario, capacità di analisi e sintesi.

*Comportamenti.* Comportamenti etici e di organizzazione del lavoro.

## **Descrizione**

Il corso intende fornire una conoscenza sufficientemente approfondita:

- dal punto di vista metodologico, dei concetti e delle tecniche fondamentali per lo sviluppo e la valutazione di applicazioni parallele ad alte prestazioni,
- dal punto di vista tecnologico, delle principali architetture parallele e distribuite ad alte

prestazioni: multiprocessor a memoria condivisa (SMP, NUMA), multicomputer, sistemi di tipo cluster di PC/workstation, con particolare enfasi sulle piattaforme abilitanti più diffuse, anche a livello commerciale, e previste nel breve/medio/lungo termine.

### **Indicazioni metodologiche**

Processo di apprendimento organizzato in moduli sequenziali e interrelati.

Verifiche intermedie continue.

Collegamenti agli altri corsi della Laurea specialistica ed alle teamtiche di ricerca.

### **Programma**

1. Metodologie di strutturazione e valutazione di sistemi e computazioni in parallelo
2. Architettura CPU pipeline ad alte prestazioni
3. Spazi di indirizzamento e strutturazione protetta a livello di processi
4. Architetture multiprocessor a memoria condivisa (SMP, NUMA)
5. Strutture di interconnessione complesse per architetture ad alto parallelismo
6. Meccanismi, costrutti linguistici e supporto per le comunicazioni in programmi paralleli
7. Piattaforme distribuite ad alte prestazioni, network computers e cluster di PC/Workstation
8. Progetto e valutazione di programmi paralleli

Ore lezione: 30 Ore esercitazione: 30

### **Bibliografia**

Materiale didattico

- M. Vanneschi, *“Architetture Parallele e Distribuite”*. Servizio Editoriale Universitario, Università di Pisa, 2005-06, nuova edizione.
- eventuale materiale integrativo fornito direttamente attraverso la pagina Web [www.di.unipi.it/~vanneschi](http://www.di.unipi.it/~vanneschi).

### **Modalità di esame**

Scritto e orale

## Aspetti etici e sociali dell'informatica

Codice: QQ001 Crediti: 3 Semestre: 2 Sigla: ASI

### Modalità di esame

[Frequenza e procedura di valutazione studenti](#)

### **Moduli:**

#### **Aspetti etici e sociali dell'informatica - 1 modulo**

#### **Docente**

Diego Latella  
*Diego.Latella@isti.cnr.it*  
*www1.isti.cnr.it/~Latella/*  
*Tel. +39 0503152982*

#### **Obiettivi di apprendimento**

Un tipico corso in Informatica e' di solito incentrato esclusivamente sugli aspetti scientifico-tecnici della disciplina. In questo corso si intende fornire agli studenti la possibilita' per sviluppare una loro personale riflessione anche su aspetti quali l'impatto sociale e militare dell'informatica, le presupposizioni della scienza e tecnologia informatica e la natura delle loro future attivita' professionali, con particolare riferimento alla relazione fra tali presupposizioni e le decisioni (etiche) che ne conseguono.

*Conoscenze.* Limiti e rischi delle tecnologie informatiche, dinamiche di interazione nei team di progetto, responsabilita' specifica degli informatici, come codificata nei codici etico/professionali del settore.

*Capacità.* Analisi critica delle tecnologie informatiche e del loro impatto sociale

*Comportamenti.* Decision-making consapevole.

#### **Descrizione**

L'uso sempre piu' diffuso della tecnologia dell'informazione (TI) nello svolgimento di attivita' anche importanti della societa' attuale fa si' che queste e, in ultima analisi la societa' stessa, dipendano sempre di piu' da tale tecnologia.

Questa nuova forma di vulnerabilita' della societa' risulta sempre piu' evidente

dalle cronache giornaliere di attacchi a sistemi informatici da parte di hacker, o dalle analisi dei rischi legati a malfunzionamenti di sistemi informatici quali, ad esempio, il millennium bug. Inoltre, l'uso dei calcolatori pone nuovi problemi legati all'utilizzo di enormi quantità di dati che investono la sfera della riservatezza dei cittadini e della sua tutela. A tali rischi si devono aggiungere quelli comunemente indicati come terrorismo informatico e guerra informatica. Nel primo caso si tratta di attacchi a infrastrutture basate su calcolatori da parte di gruppi terroristici. Nel secondo caso tali attacchi verrebbero attuati da stati sovrani attraverso loro forze militari impegnate in azioni di guerra, anche utilizzando strumenti informatici.

In quest'ultimo caso si parla di "Information Warfare" che, secondo il Dipartimento della Difesa americano, può essere definita come quelle "azioni intraprese per raggiungere la superiorità nell'informazione agendo sull'informazione dell'avversario, sui [suoi] processi basati sull'informazione, sui [suoi] sistemi informativi e sulle [sue] reti di calcolatori, allo stesso tempo facendo leva sulla, e difendendo la, propria informazione." Il Ministero della Difesa Inglese la definisce in maniera ancora più esplicita come "l'attacco deliberato [...] e sistematico su attività di informazione critiche, che tende a sfruttare, modificare e compromettere l'informazione, e all'interruzione dei servizi."

È interessante infine notare come i problemi sopra citati sono anche oggetto di una risoluzione dell'Assemblea Generale delle Nazioni Unite.

Se da una parte, la TI espone i cittadini e la società tutta ai rischi di criminalità, terrorismo e guerra elettronica a cui si è sopra accennato, dall'altra esiste l'ulteriore rischio che, per combattere tali fenomeni e difendersi dalle loro minacce, si creino dei sistemi di sorveglianza elettronica diffusa e globale, i quali costituiscono di per sé una minaccia alla riservatezza dei cittadini. Valga per tutti l'esempio del presunto sistema Echelon, ampiamente discusso anche in sede di Parlamento Europeo. Per potere affrontare efficacemente le sfide che nascono da queste nuove forme di vulnerabilità e relativi attacchi, se da una parte è necessario migliorare sempre di più la qualità della tecnologia stessa, dall'altra è essenziale creare, fra gli studenti di informatica, una coscienza e consapevolezza di tali problematiche e delle loro implicazioni sia professionali che etiche, delle possibili soluzioni e, anche, dell'eventuale inesistenza, o incompletezza, di soluzioni tecniche.

## Programma

- Introduzione e motivazioni
- Cenni al problema dei fallimenti HW e SW
- Case study 1: "Il caso del robot Killer" di R. Epstein
- Case study 2: Il sistema PATRIOT durante la guerra del Golfo
- Case study 3: Informatica e programmi USA di sicurezza nazionale (Secure Flight, US-VISIT, EU-PNR)
- Analisi delle presupposizioni prevalenti nell'informatica;
- Discussione del loro impatto nel "decision making" etico;
- Consapevolezza dei diversi fattori che possono giocare un ruolo nelle decisioni etiche.
- Codici professionali (ACM, IEEECS, AICA)

## Bibliografia

- R. Anderson, D. Johnson, D. Gotterbarn, J. Perrolle. Using the New ACM Code of Ethics in Decision Making Communications of the ACM, Feb. 1993, Vol. 36, N. 2.
- D. Latella Il caso del robot killer. Technical Report CNUCE-B4-1998-019, Consiglio Nazionale delle Ricerche, Istituto CNUCE, 1998. Traduzione italiana di 'The case of the killer robot' di R. Epstein.
- Unione Scienziati Per Il Disarmo Il Sistema Missilistico Patriot BOLLETTINO USPID - Ott. 92 (solo la parte relativa agli aspetti informatici)

Altro materiale di supporto verra' distribuito a lezione, quando necessario

Ulteriore pagina web del corso: <http://www1.isti.cnr.it/~Latella/teaching.html>

## Aspetti etici e sociali dell'informatica - 2 modulo

### Docente

Gian Piero Siroli  
*siroli@bo.infn.it*  
*siroli.web.cern.ch/siroli/*  
Tel. 051/2095240

### Obiettivi di apprendimento

Un tipico corso in Informatica e' di solito incentrato esclusivamente sugli aspetti scientifico-tecnici della disciplina. In questo corso si intende fornire agli studenti la possibilita' per sviluppare una loro personale riflessione anche su aspetti quali l'impatto sociale e militare dell'informatica, le presupposizioni della scienza e tecnologia informatica e la natura delle loro future attivita' professionali, con particolare riferimento alla relazione fra tali presupposizioni e le decisioni (etiche) che ne conseguono.

*Conoscenze.* Domini infrastrutturali vitali, vulnerabilita' cyber-fisiche ed interdipendenze. Implicazioni per la sicurezza nazionale ed internazionale. Vulnerabilita' informatica correlata a sistemi di controllo ed infrastrutture estese.

*Capacità.* Analisi critica delle tecnologie informatiche in senso esteso con particolare riguardo ai sistemi complessi. Capacita' di auto-organizzazione e coordinamento nel lavoro di gruppo.

*Comportamenti.* Decision-making consapevole.

### Descrizione

L'uso sempre piu' diffuso della tecnologia dell'informazione (TI) nello svolgimento di attivita' anche importanti della societa' attuale fa si' che queste e, in ultima analisi la societa' stessa, dipendano sempre di piu' da tale tecnologia.

Questa nuova forma di vulnerabilità della società risulta sempre più evidente dalle cronache giornalieri di attacchi a sistemi informatici da parte di hacker, o dalle analisi dei rischi legati a malfunzionamenti di sistemi informatici quali, ad esempio, il millennium bug. Inoltre, l'uso dei calcolatori pone nuovi problemi legati all'utilizzo di enormi quantità di dati che investono la sfera della riservatezza dei cittadini e della sua tutela. A tali rischi si devono aggiungere quelli comunemente indicati come terrorismo informatico e guerra informatica. Nel primo caso si tratta di attacchi a infrastrutture basate su calcolatori da parte di gruppi terroristici. Nel secondo caso tali attacchi verrebbero attuati da stati sovrani attraverso loro forze militari impegnate in azioni di guerra, anche utilizzando strumenti informatici.

In quest'ultimo caso si parla di "Information Warfare" che, secondo il Dipartimento della Difesa americano, può essere definita come quelle "azioni intraprese per raggiungere la superiorità nell'informazione agendo sull'informazione dell'avversario, sui [suoi] processi basati sull'informazione, sui [suoi] sistemi informativi e sulle [sue] reti di calcolatori, allo stesso tempo facendo leva sulla, e difendendo la, propria informazione." Il Ministero della Difesa Inglese la definisce in maniera ancora più esplicita come "l'attacco deliberato [...] e sistematico su attività di informazione critiche, che tende a sfruttare, modificare e compromettere l'informazione, e all'interruzione dei servizi."

È interessante infine notare come i problemi sopra citati sono anche oggetto di una risoluzione dell'Assemblea Generale delle Nazioni Unite.

Se da una parte, la TI espone i cittadini e la società tutta ai rischi di criminalità, terrorismo e guerra elettronica a cui si è sopra accennato, dall'altra esiste l'ulteriore rischio che, per combattere tali fenomeni e difendersi dalle loro minacce, si creino dei sistemi di sorveglianza elettronica diffusa e globale, i quali costituiscono di per sé una minaccia alla riservatezza dei cittadini. Valga per tutti l'esempio del presunto sistema Echelon, ampiamente discusso anche in sede di Parlamento Europeo. Per potere affrontare efficacemente le sfide che nascono da queste nuove forme di vulnerabilità e relativi attacchi, se da una parte è necessario migliorare sempre di più la qualità della tecnologia stessa, dall'altra è essenziale creare, fra gli studenti di informatica, una coscienza e consapevolezza di tali problematiche e delle loro implicazioni sia professionali che etiche, delle possibili soluzioni e, anche, dell'eventuale inesistenza, o incompletezza, di soluzioni tecniche.

## **Programma**

- Information Warfare (IW): definizioni di base e contesto
- Analisi delle infrastrutture critiche e possibili vulnerabilità
- Discussione di esempi di vulnerabilità in infrastrutture; meccanismi di intrusione e di rivelazione di intrusione in sistemi informatici
- Protezione delle infrastrutture critiche e problemi connessi
- Sintesi dell'attività delle Nazioni Unite e di alcuni paesi
- Tecnologie dell'informazione e sicurezza; Revolution in Military Affairs

## **Bibliografia**

- International CIIP Handbook 2004, An Inventory and Analysis of Protection Policies in Fourteen Countries, M.Dunn & I.Wigert, Center for Security Studies, ETH Zurich, 2004
- Report of the President's Commission on Critical Infrastructure Protection, Critical Foundations, USA, 1997
- National security and the Internet, G.Chapman, ISOC Geneva, 1998

Ulteriore pagina web del corso: <http://siroli.home.cern.ch/siroli/default.htm>

---

## Aspetti giuridici e professionali dell'informatica

Codice: NN001 Crediti: 3 Semestre: 1 Sigla: AGP

Settore disciplinare: IUS/04 - Diritto Commerciale

### Docente

Caterina Flick  
*flick@di.unipi.it*  
*www.di.unipi.it/~flick*  
*stanza n.297, dip.to di Informatica*  
*tel: 0502212784*

### Obiettivi di apprendimento

Conoscenza delle leggi e delle norme che regolano e tutelano la professione dell'informatico, con particolare riguardo alle fattispecie introdotte dal legislatore per descrivere e regolamentare fenomeni nuovi legati all'uso delle tecnologie dell'informazione e della comunicazione

### Descrizione

Il corso - dopo avere introdotto alcune nozioni giuridiche fondamentali - affronterà i temi connessi con diritti e doveri dell'informatico, in particolare per ciò che riguarda la creazione e la gestione del software e il mondo internet

Durante le lezioni verrà fornito materiale didattico di supporto e i riferimenti normativi fondamentali

### Programma

- Principi generali
- Software e diritto d'autore
- Tutela della privacy
- Reati informatici
- Contratti informatici

- Documento informatico e firma digitale
- Internet
- Commercio elettronico
- Informatizzazione della Pubblica Amministrazione
- Telelavoro

## Bibliografia

- I problemi giuridici di internet – a cura di E. Tosi – ed. Giuffrè 2003
- Manuale del diritto in internet – M. Sala, M. Vincenti – ed. La Tribuna 2003
- Diritto delle nuove tecnologie informatiche e dell'internet – a cura di G. Cassano – ed. IPSOA 2002
- Codice del diritto dell'informatica e di internet – ed. La Tribuna 2005
- Profili penali del lavoro on-line - a cura di E. Calmieri - ed. CEDAM 2004
- Commercio elettronico, documento informatico e firma digitale. La nuova disciplina - a cura di C. Rossello, G. Finocchiaro, E. Tosi - ed. Giappichelli 2003
- I contratti di informatica - S. Zeravscheck, E. Magni - IPSOA 2001
- Responsabilità penali e internet - C. Parodi, A. Calice - ed. Il Sole 24 ore 2001
- Lo "spamming" tra l'impostazione europea e quella statunitense - F. Giardini - in *Rassegna giuridica dell'energia elettrica* Giuffrè 2004

## Modalità di esame

La preparazione degli studenti che frequentano almeno l'80% delle lezioni verrà valutata mediante prove in itinere da effettuarsi durante lo svolgimento del corso.

Ulteriore pagina web del corso: <http://www.di.unipi.it/~flick/>

---

## Basi di dati A

Codice: AA016   Crediti: 6   Semestre: 1   Sigla: BD

Settore disciplinare:      INF/01 - Informatica

### Docente

Gualtiero Leoni  
*leoni@di.unipi.it*  
*www.di.unipi.it/~leoni*  
*stanza n.324, dip.to di Informatica*  
*tel:0502212729*

### Prerequisiti

Programmazione. Nozioni di sistemi operativi: memoria persistente, file system. Nozioni di logica matematica: dualità di De Morgan.

### **Obiettivi di apprendimento**

Il corso presenta le conoscenze di base sulle funzionalità dei DBMS, sulla progettazione di basi di dati, sul modello dei dati relazionali e sull'uso del linguaggio SQL per la definizione e uso di basi di dati.

*Conoscenze.* Conoscenze di base sulle funzionalità dei DBMS, sulla progettazione di basi di dati, sul modello dei dati relazionali e sull'uso del linguaggio SQL per la definizione e uso di basi di dati.

*Capacità.* Progettare una base di dati, esprimere interrogazioni nel linguaggio SQL, estrarre le dipendenze funzionali da una descrizione informale, applicare gli algoritmi di base della teoria relazionale, descrivere gli algoritmi e le strutture dati usate per la realizzazione delle funzionalità di base di un DBMS.

*Comportamenti.* Effettuare la raccolta di requisiti e la progettazione di una base di dati. Valutare i vantaggi e gli svantaggi dell'adozione della tecnologia delle basi di dati in una situazione applicativa.

### **Descrizione**

Gli elaboratori elettronici sono usati, nelle organizzazioni di ogni tipo e dimensione, principalmente per agevolare e potenziare le possibilità di archiviazione, recupero ed elaborazione di informazioni. I sistemi dedicati a questo scopo sono in genere basati sulla tecnologia delle basi di dati. Obiettivo del corso è di presentare le caratteristiche di questi sistemi, in particolare quelli relazionali, la loro architettura e i principi ai quali si ispirano, ponendosi principalmente dal punto di vista dei progettisti delle applicazioni.

### **Indicazioni metodologiche**

Per conseguire gli obiettivi indicati, sarà necessario:

- mantenere per tutta la durata del corso una costante attenzione al collegamento tra le nozioni presentate e la realtà del processo di progettazione e realizzazione delle basi di dati in ambito commerciale
- effettuare, all'inizio di ogni modulo, una presentazione che ne spieghi il rapporto con i moduli che lo precedono e lo seguono
- effettuare esercitazioni che siano mirate alle specifiche conoscenze e capacità che saranno verificate in sede di esame
- aiutare gli studenti a cogliere i differenti approcci da porre nello studio delle parti di natura più ingegneristica e di quelle di natura più matematica del corso

### **Programma**

Ogni punto del programma è accompagnato dal numero di ore di lezione e di esercitazione previste, per un totale di 28-20 ore rispettivamente.

- I sistemi informativi e informatici. Funzionalità dei sistemi per la gestione

di basi di dati (DBMS). (4h)

- I meccanismi di astrazione dei modelli dei dati a oggetti. La progettazione di basi di dati usando il modello a oggetti (5-5h).
- Il modello dei dati relazionale. La trasformazione di schemi a oggetti in schemi relazionali (1-1h).
- Il linguaggio SQL per creare e usare basi di dati. Interrogazioni semplici, giunzioni, quantificazioni esistenziali ed universali, raggruppamento (6-6h).
- La teoria relazionale delle basi di dati. Le dipendenze fra i dati. Decomposizioni di schemi relazionali. Forme normali. (6-6h).
- Architettura dei DBMS. Cenni alle funzionalità dei moduli per la gestione dei dati, delle transazioni e delle interrogazioni. (6-2h).

## Bibliografia

- Libro di testo: A. Albano, G. Ghelli e R. Orsini, Fondamenti di Basi di Dati, Zanichelli, Bologna, 2005.
- Lucidi del corso.
- Esercizi svolti.

## Modalità di esame

Prove di verifica intermedie, esame scritto per il recupero delle prove non superate, esame orale.

Ulteriore pagina web del corso: <http://www.di.unipi.it/~ghelli/bd1/lucidi.html>

---

## Basi di dati B

Codice: AA016   Crediti: 6   Semestre: 1   Sigla: BD

Settore disciplinare:      INF/01 - Informatica

### Docente

Giorgio Ghelli  
*ghelli@di.unipi.it*  
*www.di.unipi.it/~ghelli*  
*stanza n.290, dip.to di Informatica*  
*tel: 0502212780*

### Prerequisiti

Fondamenti di Programmazione. Linguaggio e metodi della matematica. Laboratorio di introduzione alla programmazione. Laboratorio di programmazione di strutture dati. Sistemi Operativi. Laboratorio di

programmazione concorrente e di sistema

### **Obiettivi di apprendimento**

Il corso presenta le conoscenze di base sulle funzionalità dei DBMS, sulla progettazione di basi di dati, sul modello dei dati relazionali e sull'uso del linguaggio SQL per la definizione e uso di basi di dati.

*Conoscenze.* Conoscenze di base sulle funzionalità dei DBMS, sulla progettazione di basi di dati, sul modello dei dati relazionali e sull'uso del linguaggio SQL per la definizione e uso di basi di dati.

*Capacità.* Progettare una base di dati, esprimere interrogazioni nel linguaggio SQL, estrarre le dipendenze funzionali da una descrizione informale, applicare gli algoritmi di base della teoria relazionale, descrivere gli algoritmi e le strutture dati usate per la realizzazione delle funzionalità di base di un DBMS.

*Comportamenti.* Effettuare la raccolta di requisiti e la progettazione di una base di dati. Valutare i vantaggi e gli svantaggi dell'adozione della tecnologia delle basi di dati in una situazione applicativa.

### **Descrizione**

Gli elaboratori elettronici sono usati, nelle organizzazioni di ogni tipo e dimensione, principalmente per agevolare e potenziare le possibilità di archiviazione, recupero ed elaborazione di informazioni. I sistemi dedicati a questo scopo sono in genere basati sulla tecnologia delle basi di dati. Obiettivo del corso è di presentare le caratteristiche di questi sistemi, in particolare quelli relazionali, la loro architettura e i principi ai quali si ispirano, ponendosi principalmente dal punto di vista dei progettisti delle applicazioni.

### **Indicazioni metodologiche**

Per conseguire gli obiettivi indicati, sarà necessario:

- mantenere per tutta la durata del corso una costante attenzione al collegamento tra le nozioni presentate e la realtà del processo di progettazione e realizzazione delle basi di dati in ambito commerciale
- effettuare, all'inizio di ogni modulo, una presentazione che ne spieghi il rapporto con i moduli che lo precedono e lo seguono
- effettuare esercitazioni che siano mirate alle specifiche conoscenze e capacità che saranno verificate in sede di esame
- aiutare gli studenti a cogliere i differenti approcci da porre nello studio delle parti di natura più ingegneristica e di quelle di natura più matematica del corso

### **Programma**

Ogni punto del programma è accompagnato dal numero di ore di lezione e di esercitazione previste, per un totale di 28-20 ore rispettivamente.

- I sistemi informativi e informatici. Funzionalità dei sistemi per la gestione di basi di dati (DBMS). (4h)

- I meccanismi di astrazione dei modelli dei dati a oggetti. La progettazione di basi di dati usando il modello a oggetti (5-5h).
- Il modello dei dati relazionale. La trasformazione di schemi a oggetti in schemi relazionali (1-1h).
- Il linguaggio SQL per creare e usare basi di dati. Interrogazioni semplici, giunzioni, quantificazioni esistenziali ed universali, raggruppamento (6-6h).
- La teoria relazionale delle basi di dati. Le dipendenze fra i dati. Decomposizioni di schemi relazionali. Forme normali. (6-6h).
- Architettura dei DBMS. Cenni alle funzionalità dei moduli per la gestione dei dati, delle transazioni e delle interrogazioni. (6-2h).

Ore lezione: 28 Ore esercitazione: 20

### Bibliografia

- Libro di testo: A. Albano, G. Ghelli e R. Orsini, Fondamenti di Basi di Dati, Zanichelli, Bologna, 2005.
- Lucidi del corso.
- Esercizi svolti.

### Modalità di esame

Prove di verifica intermedie, esame scritto per il recupero delle prove non superate, esame orale.

Ulteriore pagina web del corso: <http://www.di.unipi.it/~ghelli/bd1/lucidi.html>

---

## Basi di dati: laboratorio

Codice: AA023 Crediti: 6 Semestre: 2 Sigla: BDL

Settore disciplinare: INF/01 - Informatica

### Docente

Giorgio Ghelli  
*ghelli@di.unipi.it*  
*www.di.unipi.it/~ghelli*  
*stanza n.290, dip.to di Informatica*  
*tel:0502212780*

### Prerequisiti

Basi di Dati.

### Obiettivi di apprendimento

Completare le nozioni possedute sulla progettazione e realizzazione di

applicazioni su basi di dati.

*Conoscenze.* Metodi per la progettazione di basi di dati. Linguaggi per la costruzione di applicazioni su basi di dati.

*Capacità.* Raccolta di requisiti. Progettazione e realizzazione di una piccola applicazione su basi di dati.

*Comportamenti.* Sapere lavorare in gruppo.

## **Descrizione**

In questo corso si approfondiscono le nozioni relative ad analisi e progettazione di dati, procedure ed interfacce di applicazioni per basi di dati, ed alla loro realizzazione, introdotte nel corso di Basi di Dati e Sistemi Informativi 1. Tutte le nozioni introdotte vengono immediatamente sperimentate dagli studenti, utilizzando notazioni standard ed un sistema commerciale, quale ad esempio ORACLE DBMS ed ORACLE WebServer, sviluppando un case study che si conclude, alla fine del corso, con la realizzazione da parte degli studenti di un sistema funzionante.

## **Indicazioni metodologiche**

La didattica del corso è impostata sull'acquisizione di conoscenze e capacità attraverso la sperimentazione, e sul lavoro di gruppo.

## **Programma**

In questo corso si approfondiscono le nozioni relative ad analisi e progettazione di dati, procedure ed interfacce di applicazioni per basi di dati, ed alla loro realizzazione, introdotte nel corso di Basi di Dati e Sistemi Informativi 1. Tutte le nozioni introdotte vengono immediatamente sperimentate dagli studenti, utilizzando notazioni standard ed un sistema commerciale, quale ad esempio ORACLE DBMS ed ORACLE WebServer, sviluppando un case study che si conclude, alla fine del corso, con la realizzazione da parte degli studenti di un sistema funzionante.

Il corso inizia con la presentazione di un problema applicativo agli studenti, attraverso interviste. Gli studenti, divisi in piccoli gruppi, preparano una specifica dei requisiti. Le diverse specifiche vengono valutate, ed una viene scelta come la specifica ufficiale.

Agli studenti viene presentato il sistema ORACLE, il linguaggio PL/SQL, ed un'API per la costruzione di interfacce, basata su HTML. Le funzionalità definite nella specifica dei requisiti vengono distribuite tra gli studenti, divisi nuovamente in piccoli gruppi, per essere realizzate con gli strumenti visti durante il corso. Gli studenti preparano anche uno schema SQL comune sul quale si innesteranno tutti i pezzi di codice da loro prodotti.

Il giorno dell'esame tutti i pezzi di codice prodotti vengono integrati in un unico sistema, e valutati separatamente per la valutazione finale degli studenti.

Ore laboratorio: 48

## **Bibliografia**

Libro di testo: A. Albano, G. Ghelli e R. Orsini, Fondamenti di Basi di Dati, Zanichelli, Bologna, 2005. Manuali ORACLE

Testi di consultazione

C. Batini, S. Ceri, S. Navathe, Conceptual database design Q An Entity Relationship Approach, The Benjamin/Cummings Publishing Company, Inc., Redwood City, California. 1992.

### Modalità di esame

Valutazione e discussione di due progetti realizzati durante lo svolgimento del corso.

Ulteriore pagina web del corso: <http://projects.cli.di.unipi.it/bdl2004/>

---

## Basi di dati: strutture e algoritmi

Codice: AA022   Crediti: 6   Semestre: 1   Sigla: BSA

Settore disciplinare:            INF/01 - Informatica

### Docente

Antonio Albano  
*albano@di.unipi.it*  
*www.di.unipi.it/~albano*  
*stanza n.291, dip.to di Informatica*  
*tel: 0502212769*

### Prerequisiti

Conoscenza dei DBMS relazionali e del linguaggio SQL. Raccomandato "Basi di dati"

### Obiettivi di apprendimento

Il corso presenta le tecniche per la realizzazione di sistemi per la gestione di basi di dati relazionali. Vengono discusse le funzionalità, le strutture e gli algoritmi dei moduli principali per (a) organizzare i dati nella memoria permanente, (b) per ottimizzare l'esecuzione delle operazioni sui dati espresse nel linguaggio SQL e (c) per gestire le transazioni e gli accessi concorrenti ai dati. Il corso presenta anche un approccio alla progettazione fisica e alla messa punto di basi dati per garantire le prestazioni attese.

*Conoscenze.* Lo studente acquisirà conoscenze di base, sia sui metodi che sulle tecnologie, per affrontare i problemi che si presentano quando occorre realizzare sistemi per la gestione di dati e la progettazione fisica di basi di dati relazionali

*Capacità.* Lo studente saprà progettare e realizzare (a) strutture di memorizzazione per dati persistenti, (b) algoritmi per ottimizzare l'esecuzione delle interrogazioni, (c) algoritmi per gestire transazioni e concorrenza. Lo studente saprà inoltre progettare gli aspetti fisici delle basi di dati, valutare la tecnologia dei prodotti commerciali disponibili sulla base della piena comprensione delle soluzioni tecniche utilizzate e comprendere le innovazioni tecnologiche del settore.

*Comportamenti.* Lo studente sarà essere indipendente nella valutazione degli strumenti di gestione di basi di dati, nella valutazione della tecnologia e nella comunicazione con gli esperti dei domini applicativi e tecnologici per la progettazione di basi di dati. Lo studente sarà essere inoltre consapevole delle problematiche della sicurezza.

## Descrizione

*Vengono presentate le tecniche per la realizzazione di sistemi per la gestione di basi di dati, in particolare quelli relazionali. Vengono discusse le funzionalità, le strutture e gli algoritmi dei moduli principali per (a) organizzare i dati nella memoria permanente, (b) per ottimizzare l'esecuzione delle operazioni sui dati espresse nel linguaggio SQL e (c) per gestire le transazioni e gli accessi concorrenti ai dati.*

## Programma

- *Introduzione (2 ore)* Introduzione, obiettivi del corso. Architettura dei sistemi per basi di dati I moduli funzionali.
- *I moduli del gestore della memoria relazionale per la gestione dei dati (14 ore + 6 ore di esercizi)* Il gestore della memoria permanente e il gestore del buffer. Il gestore delle strutture di memorizzazione: Organizzazioni seriali e sequenziali; organizzazioni per chiavi primarie e secondarie; organizzazioni per associazioni. Il gestore dei metodi di accesso: l'interfaccia della macchina fisica.  
  
*I moduli del gestore della macchina relazionale (8 ore + 6 ore di esercizi)* Il gestore del catalogo. Il gestore delle autorizzazioni. Il gestore delle interrogazioni: algoritmi per realizzare gli operatori dell'algebra relazionale e per ottimizzare l'esecuzione delle interrogazioni in SQL. L'esecutore dei piani di accesso.
- *I moduli del gestore della memoria relazionale per la gestione delle transazione (6 ore + 4 ore di esercizi)* Il gestore dell'affidabilità e il gestore della concorrenza: tecniche per la gestione delle transazioni e della concorrenza.
- *Progettazione fisica e messa a punto di basi di dati (2 ore)*

Ore lezione: 32 Ore esercitazione: 16

## Bibliografia

- Appunti delle lezioni.
- Raccolta di esercizi e di soluzioni di prove d'esame.
- Testo di riferimento: *A. Albano, Costruire sistemi per basi di dati*, Addison-Wesley, Milano, 2001.

I libri consigliati sono alla pagina web del corso.

### Modalità di esame

Una prova orale preceduta da una prova scritta della durata di un'ora.

Ulteriore pagina web del corso:

<http://www.di.unipi.it/~albano/corsi/ProgrammaBD2.html>

---

## Calcolabilità e complessità

Codice: AA025   Crediti: 9   Semestre: 1   Sigla: CC

Settore disciplinare:            INF/01 - Informatica

### Docente

Pierpaolo Degano  
*degano@di.unipi.it*  
*www.di.unipi.it/~degano*  
*stanza n.285, dip.to di Informatica*  
*tel: 0502212757*

### Prerequisiti

Un po' di teoria degli insiemi e delle strutture algebriche; un po' di logica; la nozione di funzione.

### Obiettivi di apprendimento

Comprendere quali sono i problemi risolvibili meccanicamente, in assenza e in presenza di vincoli sulle risorse di calcolo

*Conoscenze.* Conoscenze di base

*Capacità.* Capacità di comprensione, apprendimento e ragionamento logico-deduttivo

*Comportamenti.* Comportamenti razionali

### Descrizione

Il corso introduce le nozioni fondamentali della teoria della calcolabilità e della complessità. La prima parte delinea i concetti e la natura dei problemi che hanno soluzione effettiva. La seconda parte caratterizza i problemi che sono

risolvibili con risorse di calcolo limitate.

- Macchine di Turing (deterministiche e non, a più nastri, I/O)
- Linguaggi calcolabili, MdT universale
- Funzioni ricorsive e linguaggi di programmazione,
- Totalità e diagonalizzazione
- Riducibilità, problemi insolubili
- Funzioni di misura di tempo e spazio
- Classi (tempo/spazio) deterministiche e non, P- e NP-completezza
- Altre classi (co-NP, caso, approssimazione, parallelismo)

### Indicazioni metodologiche

Organizzare il processo di apprendimento in sequenza logica

### Programma

- Turing machines (deterministic, non-deterministic, k-tapes, I/O)
- Computable languages, Universal Turing machine
- Recursive functions and programming languages
- Total functions and diagonalization
- Reducibility, unsolvable problems
- Time and space measures
- Deterministic and non-deterministic complexity classes (time/space), P- e NP-completeness
- Other classes (co-NP, random, approximation, parallelism)

Ore lezione: 50

Ore esercitazione: 15

Ore laboratorio: 0

Ore  
seminari: 0

### Bibliografia

- Ch.H. Papadimitriou, Computational Complexity, Addison-Wesley, 1994.  
R.J. Soare, Recursively Enumerable Sets and Degrees, Springer-Verlag, 1988.  
P. Degano, Notes.  
E. Börger, Computability, Complexity, Logic, North-Holland, 1989.  
A. Bernasconi, B. Codenotti, Introduzione alla Complessità Computazionale, Springer, 1998.  
T.H. Cormen, C.E. Leiserson, L.R. Rivest, Introduction to Algorithms, MIT Press, 1990.  
M.R. Garey, D.S. Johnson, Computers and Intractability, Freeman & Co., 1979.  
H.R. Lewis, Ch.H. Papadimitriou, Elements of the Theory of Computation, Prentice-Hall, 1981.  
N.D. Jones, Computability and Complexity, MIT Press, 1997.  
R. Sommerhalden, S.C. van Westrhenen, The Theory of Computability, Addison-Wesley, 1988.

### Modalità di esame

Scritto e orale

Ulteriore pagina web del corso:

[http://www.di.unipi.it/segreteria/corsi/CorsiNuovi/AA025\\_unico.html](http://www.di.unipi.it/segreteria/corsi/CorsiNuovi/AA025_unico.html)

---

## Calcolo della probabilità e statistica A

Codice: AA009   Crediti: 6   Semestre: 2   Sigla: CPS

Settore disciplinare:            MAT/06 - Probabilità e  
Statistica Matematica

### Docente

Rita Giuliano

*giuliano@dm.unipi.it*

*www.dm.unipi.it/~giuliano/*

*Tel. 050-2213242*

### Prerequisiti

Si richiede la conoscenza dell'Analisi Matematica quale è fornita dal corso relativo.

### Obiettivi di apprendimento

Fornire elementi di Statistica Descrittiva, Calcolo delle Probabilità e Statistica Inferenziale.

### Descrizione

Il corso intende fornire i primi elementi di Calcolo delle Probabilità e Statistica. Il corso è costituito di 40 ore, 25 delle quali di lezioni teoriche e 15 di esercitazioni in aula.

### Indicazioni metodologiche

lezione frontale ed esercitazione in aula.

### Programma

STATISTICA DESCRITTIVA.

Organizzazione e descrizione dei dati. Caratteri: non numerici, numerici, discreti, continui; modalità, effettivi, frequenze relative, frequenze cumulative. Tabelle e grafici delle frequenze. Raggruppamento dei dati, istogrammi, ogive e diagrammi stem and leaf. Le grandezze che sintetizzano i dati. Media, mediana, e moda. Varianza e deviazione standard. Quantili, quartili, percentili, range, intervallo interquartile, box plot.

Insiemi di dati bivariati, diagramma di dispersione, covarianza, coefficiente di correlazione, retta di regressione  $\hat{y}$  dei minimi quadrati), interpretazione.

## PROBABILITÀ.

Spazi di probabilità e loro proprietà. Probabilità condizionale, indipendenza di eventi, formula di Bayes. Spazi di probabilità uniformi, elementi di calcolo combinatorio.

Schema delle prove indipendenti (o di Bernoulli).

Concetto di variabile aleatoria. Variabili aleatorie discrete. Densità discrete e proprietà.

Definizione di funzione di ripartizione e proprietà.

Principali densità discrete: Bernoulliana, binomiale, ipergeometrica, geometrica, di Poisson.

Vettori aleatori discreti. Densità congiunte e marginali. Variabili aleatorie discrete indipendenti.

Definizione di speranza matematica per v. a. discrete e proprietà. Varianza, covarianza e proprietà.

V. a. assolutamente continue. Principali densità assolutamente continue: uniforme, esponenziale, normale, chi quadrato, di Student.

## STATISTICA

Concetto di stimatore. Stimatori corretti, rischio quadratico, stimatori dei momenti, stimatori di massima verosimiglianza.

Definizione di quantile. Uso delle tavole dei quantili della  $N(0,1)$ , della chi-quadrato e della  $t$  di Student.

Principali proprietà delle densità normali (media, varianza, comportamento rispetto alla somma e alle trasformazioni lineari affini). Enunciato della legge dei grandi numeri, del teorema limite centrale, del teorema di Cochran. Approssimazione normale.

Concetto di intervallo di fiducia: intervalli di fiducia per la media di una normale con varianza nota e non nota. Intervalli per una proporzione. Intervalli per la varianza. Test statistici. Generalità, livello, regione critica, errore di prima e seconda specie, funzione potenza.

Test di Student (per la media di una normale con varianza nota e non nota).

Test per una proporzione. Test di Fisher (per la varianza di un anormale). Il test del chi-quadrato.

## Bibliografia

**R. Giuliano, "Elementi di Calcolo delle Probabilità", ETS Editrice**

**R. Giuliano, "Elementi di Statistica Inferenziale", ETS Editrice**

Testi consigliati:

1) P. Baldi, "Calcolo delle Probabilità e Statistica", Mc Graw Hill

2) S. Borra - A. Di Ciaccio, "Statistica" metodologie per le scienze economiche e sociali, Mc Graw Hill

3) D. M. Levine - T. C. Krehbiel - M. L. Berenson, "Statistica", APOGEO

4) S. Ross, "Probabilità e Statistica" per l'ingegneria e le scienze, APOGEO

5) S. Ross, "Calcolo delle Probabilità", APOGEO

Note:

- Per la parte di Statistica Descrittiva, si consiglia di utilizzare il testo 4.
- Il testo 4 contiene anche molti esercizi (non risolti).
- A lezione verranno distribuite fotocopie di esercizi risolti.
- Altri esercizi, in parte risolti, sono a disposizione sul sito della [Prof. M. De Donno](#).

### Modalità di esame

Una prova scritta e una orale. Per l'ammissione all'orale si richiede una votazione di almeno 15/30 nello scritto. Una votazione superiore o uguale a 27/30 nello scritto esonera dalla prova orale. Il superamento delle prove in itinere esonera dalla prova scritta solo per la sessione di esami immediatamente successiva al termine del corso.

Ulteriore pagina web del corso: <http://www.dm.unipi.it/~giuliano>

---

## Calcolo della probabilità e statistica B

Codice: AA009   Crediti: 6   Semestre: 2   Sigla: CPS

Settore disciplinare:            MAT/06 - Probabilità e  
Statistica Matematica

### Docente

Marzia De Donno  
*mdedonno@dm.unipi.it*  
*www.dm.unipi.it/~mdedonno/*  
*Tel. 050-2213317*

### Prerequisiti

Elementi di base di teoria degli insiemi e calcolo combinatorio.  
Elementi di analisi: funzione logaritmica ed esponenziale, limiti, serie, derivate ed integrali.

### Bibliografia

### Modalità di esame

L'esame consiste in una prova scritta ed una prova orale.

Ulteriore pagina web del corso:

<http://www.dm.unipi.it/~mdedonno/teaching/aa0506/PIcps/cps200506pi.html>

---

# Calcolo numerico A

Codice: AA010   Crediti: 6   Semestre: 1   Sigla: CN

Settore disciplinare:            MAT/08 - Analisi Numerica

## Docente

Ornella Menchi  
*menchi@di.unipi.it*  
*stanza n.328, dip.to di Informatica*  
*tel:0502212740*

## Prerequisiti

E' raccomandato il superamento degli esami di Analisi e Algebra

## Obiettivi di apprendimento

Il corso propone i principali problemi del calcolo scientifico, e introduce lo studente all'apprendimento, l'uso e la valutazione delle metodologie di risoluzione di tali problemi. Particolare attenzione è rivolta agli aspetti computazionali, quali il condizionamento dei problemi esaminati, la stabilità e la complessità dei metodi proposti.

*Conoscenze.* Lo studente apprende i metodi di risoluzione dei piu' comuni problemi del calcolo numerico, unitamente alle proprietà matematiche dei modelli usati e quelle numeriche degli algoritmi proposti

*Capacità.* Lo studente viene messo in condizione di affrontare un problema di calcolo, ovvero di usare un metodo di risoluzione e di valutare il comportamento numerico dell'algoritmo che lo implementa

*Comportamenti.* Non è fra gli obiettivi del corso quello di indicare comportamenti o modelli deontologici.

## Descrizione

Il corso presenta i metodi fondamentali per risolvere numericamente i problemi matematici, mettendo in evidenza gli aspetti computazionali quali il condizionamento, la propagazione degli errori e la complessità.

## Programma

- Rappresentazione dei numeri reali, analisi dell'errore, complessità numerica.
- Metodi iterativi per equazioni non lineari.
- Metodi numerici per l'algebra lineare.
- Interpolazione e integrazione numerica.

Ore lezione: 32   Ore esercitazione: 16

## Bibliografia

Dispense del corso stampate dal SEU.

Testi complementari: R. Bevilacqua, D. Bini, M. Capovani, O. Menchi, Introduzione alla matematica Computazionale, Zanichelli. D. Bini, M. Capovani, O. Menchi, Metodi numerici per l'algebra lineare, Zanichelli. R. Bevilacqua, D. Bini, M. Capovani, O. Menchi, Metodi numerici, Zanichelli.

## Modalità di esame

Due prove scritte intermedie di verifica.

Esame finale con prova scritta e orale, con esenzione dalla prova scritta nel caso di superamento delle verifiche intermedie.

Ulteriore pagina web del corso: <http://www.di.unipi.it/~delcorso/cnum.html>

---

## Calcolo numerico B

Codice: AA010   Crediti: 6   Semestre: 1   Sigla: CN

Settore disciplinare:            MAT/08 - Analisi Numerica

### Docente

Roberto Bevilacqua  
*bevilacq@di.unipi.it*  
*stanza n. 370, dip.to di Informatica*  
*tel: 0502212795*

### Prerequisiti

E' raccomandato il superamento degli esami di Analisi e Algebra.

### Obiettivi di apprendimento

Il corso propone i principali problemi del calcolo scientifico, e introduce lo studente all'apprendimento, l'uso e la valutazione delle metodologie di risoluzione di tali problemi. Particolare attenzione è rivolta agli aspetti computazionali, quali il condizionamento dei problemi esaminati, la stabilità e la complessità dei metodi proposti.

*Conoscenze.* Lo studente apprende i metodi di risoluzione dei piu' comuni problemi del calcolo numerico, unitamente alle proprietà matematiche dei modelli usati e quelle numeriche degli algoritmi proposti.

*Capacità.* Lo studente viene messo in condizione di affrontare un problema di calcolo, ovvero di usare un metodo di risoluzione e di valutare il comportamento numerico dell'algoritmo che lo implementa.

*Comportamenti.* Non è fra gli obiettivi del corso quello di indicare comportamenti o modelli deontologici.

### **Descrizione**

Il corso presenta i metodi fondamentali per risolvere numericamente i problemi matematici, mettendo in evidenza gli aspetti computazionali quali il condizionamento, la propagazione degli errori e la complessità.

### **Programma**

- Rappresentazione dei numeri reali, analisi dell'errore, complessità numerica.
- Metodi iterativi per equazioni non lineari.
- Metodi numerici per l'algebra lineare.
- Interpolazione e integrazione numerica.

Ore lezione: 32 Ore esercitazione: 16

### **Bibliografia**

Dispense del corso stampate dal SEU.

Testi complementari: R. Bevilacqua, D. Bini, M. Capovani, O. Menchi, Introduzione alla matematica Computazionale, Zanichelli. D. Bini, M. Capovani, O. Menchi, Metodi numerici per l'algebra lineare, Zanichelli. R. Bevilacqua, D. Bini, M. Capovani, O. Menchi, Metodi numerici, Zanichelli.

### **Modalità di esame**

Due prove scritte intermedie di verifica.

Esame finale con prova scritta e orale, con esenzione dalla prova scritta nel caso di superamento delle verifiche intermedie.

Ulteriore pagina web del corso: <http://www.di.unipi.it/~delcorso/cnum.html>

---

## **Compilatori**

Codice: AA026 Crediti: 6 Semestre: 2 Sigla: COP

Settore disciplinare: INF/01 - Informatica

### **Docenti**

Andrea Maggiolo Schettini  
*maggiolo@di.unipi.it*  
*www.di.unipi.it/~maggiolo*  
*stanza n.279, dip.to di Informatica*  
*tel: 0502212759*  
Marco Bellia

*bellia@di.unipi.it*  
*www.di.unipi.it/~bellia*  
*stanza n.371, dip.to di Informatica*  
*tel: 0502212781*

## **Prerequisiti**

*1 modulo:* Fondamenti di programmazione.

*2 modulo:* Fondamenti di programmazione, Algoritmica e laboratori I e II.

## **Obiettivi di apprendimento**

*1 modulo:* La conoscenza dei concetti della teoria degli automi e dei linguaggi formali di base e il loro utilizzo per la compilazione dei linguaggi di programmazione.

*2 modulo:* Introdurre le principali problematiche inerenti la realizzazione di traduttori e interpreti per linguaggi di programmazione. Introdurre le tecniche e gli strumenti per analisi statica, traduzione, generazione e interpretazione di codice.

## **Descrizione**

*1 modulo:* La teoria degli automi e dei linguaggi formali è alla base della descrizione dei linguaggi di programmazione, della costruzione dei loro riconoscitori e traduttori, della realizzazione di strumenti di elaborazione testuale. Il corso esporrà la teoria degli automi finiti e dei linguaggi non contestuali e descriverà algoritmi e strutture dati per l'analisi lessicale e l'analisi sintattica dei programmi.

*2 modulo:* Si introducono tecniche basate su grammatiche ad attributi e su syntax-directed definitions per costruire analizzatori semantici e traduttori guidati dalla sintassi. Queste tecniche sono utilizzate in contesti differenti e nella costruzione di strumenti quali riconoscitori di linguaggi non context-free, esecutori di linguaggi non-standard, editor guidati dalla sintassi, sistemi per la composizione grafica di testi e immagini.

## **Programma**

*1 modulo (3 cfu) - prof. A.Maggiolo-Schettini:*

- Automi a stati finiti. Espressioni regolari. Proprietà degli insiemi regolari. Analisi lessicale. (6 ore)
- Grammatiche non contestuali. Automi a pila. Proprietà dei linguaggi non-contestuali. (6 ore)
- Grammatiche LLK e LRK. Analizzatori sintattici ascendenti e discendenti. (8 ore)
- Generatori di analizzatori lessicali e sintattici. (4 ore)

*Pagina web:* <http://www.di.unipi.it/~maggiolo/lucidi-COP.html>

*2 modulo (3 cfu) - prof. M. Bellia:*

- Attributi, grammatiche L-attributate, esecutori top-down e bottom-up, rappresentazioni di programmi (ore 6)
- Analisi statica: proprietà statiche di programmi, sistemi di tipi e controllo di tipi (ore 6)
- Macchine astratte: supporto run time, memoria statica e dinamica, tabella dei simboli (ore 4)
- Generazione di codice: linguaggi intermedi, traduzione di espressioni, controllo di sequenza, astrazioni procedurali (8 ore)

Pagina web: <http://www.di.unipi.it/~bellia>

## Bibliografia

### 1 modulo:

- Hopcroft, J.E., Ullman J.D., *Introduction to Automata Theory, Languages, and Computation*, Addison Wesley, Reading, Mass., 1979.
- Hopcroft, J.E., Motwani, R., Ullman, J.D., *Automi, linguaggi e calcolabilità*, Addison Wesley, Pearson Education Italia, 2003.
- Salomaa, A., *Formal Languages*, Academic Press, New York, 1987.
- Aho A.V., Sethi R., Ullman J.D., *Compilers: Principles, Techniques and Tools*, Addison-Wesley, Reading, Mass., 1988.

### 2 modulo:

- Aho A.V., Sethi R., Ullman J.D., *Compilers. Principles, Techniques and Tools*, Addison Wesley, Reading, Mass., 1986.
- Wilhelm R. and Maurer D., *Compiler Design*, Addison-Wesley, 1996.

## Complementi di gestione di rete

Codice: AA052   Crediti: 6   Semestre: 2   Sigla: SGR

Settore disciplinare: INF/01 - Informatica

### Docente

Luca Deri  
*deri@di.unipi.it*  
*luca.ntop.org/*  
*stanza n.392, dip.to di Informatica*  
*tel:0502213105*

### Prerequisiti

Reti di Calcolatori (AA019)

### Obiettivi di apprendimento

Fornire agli studenti una conoscenza di base dei problemi della gestione di rete, degli strumenti per la gestione disponibili e dei principi di base che permettono di gestire complesse reti di comunicazioni.

*Conoscenze.* Al termine del corso lo studente sarà capace di individuare la metodologia e gli strumenti più idonei a gestire una rete di calcolatori.

*Capacità.* Lo studente sarà in grado di analizzare un problema di gestione di rete e di risolverlo utilizzando gli strumenti presentati nel corso.

*Comportamenti.* Metodologia per affrontare in maniera un problema di gestione di rete e consapevolezza dei limiti degli strumenti oggi disponibili.

## **Descrizione**

Questo corso tratta la gestione di reti dati e di telecomunicazioni. Il corso comprende una parte teorica ed una pratica.

### 1. Teoria

- o Principali standard e protocolli per la gestione di rete
- o Differenti paradigmi per la gestione di rete: descrizione e confronto.

### 2. Pratica

Oltre che a far vedere nella realtà alcuni concetti spiegati durante il corso, presenta di volta in volta i nuovi paradigmi di gestione e gli ultimi risultati della ricerca nel campo della gestione di rete. Sono inoltre illustrati i principali strumenti di gestione di rete sia Open Source che commerciali.

## **Indicazioni metodologiche**

Durante il corso saranno presentati sia strumenti per la gestione di rete che casi reali in cui verranno proposte e confrontate varie soluzioni al problema. Questo per fornire non solo una conoscenza di base, ma per permettere allo studente di rendersi conto di quali sono i problemi di gestione più comuni e come questi vengono affrontati.

## **Programma**

### 1. Teoria:

1. Introduzione
2. Internet-Management: SNMP-based Management
3. Altri paradigmi di Management
4. Attuali argomenti di ricerca nel campo della gestione di rete

### 2. Pratica:

1. Introduzione alla misurazione del traffico di Rete
2. Misure utilizzando SNMP
3. Monitoraggio di Reti Remote
4. Flow-based Measurement: NetFlow, sFlow, IPFIX
5. Misurazione di Rete: Case Studies

## **Bibliografia**

- J. Sloan, *Network Troubleshooting Tools*, O'Reilly, 2001, <http://www.oreilly.com/catalog/nettroubletools/index.html>.
- D. Mauro e K. Smith, *Essential SNMP*, 2nd Edition, O'Reilly, 2005, <http://www.oreilly.com/catalog/esnmp2/index.html>.

### Modalità di esame

Progettino ed orale.

Ulteriore pagina web del corso: <http://luca.ntop.org/>

---

## Comunicazione

Codice: LL002    Crediti: 3    Semestre: 2    Sigla: COM

Settore disciplinare: L-FIL-LET/12 - Linguistica Italiana

### Docente

Robert Lee  
*robert.lee@hotmail.it*

### Prerequisiti

Nessuno.

### Obiettivi di apprendimento

Scopo del corso è quello di introdurre lo studente alle problematiche della comunicazione, prima in forma astratta, e poi più concretamente nella comunicazione usata dai sistemi di informazione.

*Conoscenze.*

Al termine del corso lo studente sarà a conoscenza degli schemi fondamentali di comunicazione utilizzati dai mezzi di comunicazione.

*Capacità.* Lo studente sarà in grado di scrivere un comunicato stampa, ed un breve pezzo giornalistico.

*Comportamenti.* Lo studente sarà a conoscenza delle principali problematiche deontologiche che proteggono i minori e le fasce più deboli.

### Descrizione

Il corso introduce lo studente alle principali problematiche della comunicazione. Dopo aver introdotto il problema della comunicazione vengono esaminati gli strumenti impiegati nelle principali forme di comunicazione. Allo studente sarà coinvolto nel corso nella stesura di comunicati e di brevi testi giornalistici.

## Programma

- La struttura della comunicazione
- I periodici
- I giornali
- La televisione
- La radio
- Come scrivere un comunicato stampa

Ore lezione: 20

## Bibliografia

Il material del corso è prevalentemente on-line ed accessibile tramite collegamenti presenti nel sito del corso.

## Modalità di esame

L'esame consiste nella stesura di un piccolo elaborato di tipo giornalistico.

Ulteriore pagina web del corso: <http://rotor.di.unipi.it/Comunicazione/>

---

## Costruzione di interfacce

Codice: AA027   Crediti: 9   Semestre: 1   Sigla: CI

Settore disciplinare:      INF/01 - Informatica

## Docente

Giuseppe Attardi  
*attardi@di.unipi.it*  
*www.di.unipi.it/~attardi*  
*stanza n.292, dip.to di Informatica*  
*tel:0502212744*

## Prerequisiti

Laboratori del primo e secondo anno, conoscenza del linguaggio di programmazione Java.

## Obiettivi di apprendimento

Il corso introduce alle problematiche relative alla progettazione e realizzazione di applicazioni interattive in sistemi grafici basati su interfacce a finestre. Si introducono le tecniche fondamentali utilizzate per realizzare questo tipo di interfacce e i principali modelli impiegati per la realizzazione di applicazioni. Si considera la struttura dei framework per realizzare interfacce, e come possono

essere estesi con nuovi componenti. Viene mostrato come realizzare interfacce basate su Web-browser utilizzando standard come CSS e DHTML. Breve introduzione alla grafica tridimensionale, strumenti e applicazioni.

## Descrizione

Il corso introduce alle problematiche relative alla progettazione e realizzazione di applicazioni interattive in sistemi grafici basati su interfacce a finestre. Si introducono le tecniche fondamentali utilizzate per realizzare questo tipo di interfacce e i principali modelli impiegati per la realizzazione di applicazioni. Si considera la struttura dei framework per realizzare interfacce, e come possono essere estesi con nuovi componenti. Viene mostrato come realizzare interfacce basate su Web-browser utilizzando standard come CSS e DHTML. Breve introduzione alla grafica tridimensionale, strumenti e applicazioni.

## Indicazioni metodologiche

Progetto intermedio (realizzazione di un componente), progetto finale (realizzazione di un'applicazione grafica) con orale.

## Programma

- Introduzione a C++
- Sistemi grafici
- Fondamenti di grafica
- Modello model-view-controller
- Programmazione ad eventi
- Sistemi di riferimento e trasformazioni geometriche
- Librerie e framework per la grafica
- Programmazione di interfacce grafiche
- Interfacce Web
- Cenni alla grafica 3D

Ore lezione: 36 Ore esercitazione: 36

## Bibliografia

- R. Scateni, P. Cignoni, C. Montani, R. Scopigno - *Fondamenti di grafica tridimensionale interattiva*, McGraw-Hill, 2005, ISBN: 88 386 6215-0
- Fondamenti di Computer graphics G. Attardi, Università di Pisa, SEU, 1998.

Testi di consultazione:

- C. Petzold, *Programmare Windows con C#*, Microsoft Press, 2001, ISBN: 0-7356-1370-2

## Modalità di esame

Progetto intermedio (realizzazione di un componente), progetto finale (realizzazione di un'applicazione grafica) con orale.

Ulteriore pagina web del corso: <http://rotor.di.unipi.it/cisterni/CI>

---

## Cultura Europea

Codice: ZY567   Crediti: 3   Semestre: 1   Sigla: CUE

Settore disciplinare:            INF/01 - Informatica

### Docente

Gianluigi Ferrari  
*giangi@di.unipi.it*  
*www.di.unipi.it/~giangi*  
*stanza n. 358, dip.to di Informatica*  
*tel: 0502212766*

---

## PP001 Economia dei sistemi industriali (CFU:3)

mutuato da:

### Economia dei sistemi industriali

Codice: PP326   Crediti: 6   Semestre: 1   Sigla: ESI

Settore disciplinare:            SECS-P/07 - Economia  
Aziendale

### Docente

Dionisia Cazzaniga Francesetti  
*dcf@di.unipi.it*  
*stanza n. 394, dip.to di Informatica*  
*tel: 0502212711*

### Prerequisiti

nozioni analisi

### Obiettivi di apprendimento

Creare un tecnico informatico facilmente inseribile nelle imprese

manifatturiere o di servizio. Il soggetto deve essere in grado di analizzare un'impresa sulla base di cognizioni micro e macro economiche di base compreso il bilancio. Deve conoscere la formazione dei costi ed il loro controllo , nonchè come si organizza una impresa manifatturiera.

## Programma

1) Si premettono le nozioni sul bilancio; libro di testo: Caso e Gioffrè: **BILANCIO IN FORMA ABBREVIATA** ed Buffetti

2) Nozioni di base di micro e macro economia. Testo: Samuelson, XV edizione 1996—sono indicate di seguito le pagine ed i capitoli da seguire.

Domanda e offerta. pg 40-52. Elasticità della domanda rispetto al prezzo 6, pg. 0-68. Domanda e comportamento del consumatore: dall'utilità al mercato, pg. 78-87. Curva indifferenza pg. 92-95 Produzione e organizzazione delle imprese 101-109 Analisi economica dei costi 114-123 Comportamento di offerta dell'impresa concorrenziale 137-143 Concorrenza imperfetta e problema del monopolio 160-172 Calcolo del valore di un investimento o valore attuale delle attività patrimoniali 258-263 Quadro generale di macroeconomia 390-401 Il prodotto interno lordo, criterio di misura 410-415 Il consumo e l'investimento 429-443 Elementi che incidono sulla domanda aggregata e loro rappresentazione nel moltiplicatore 452-462 (Moneta ed attività bancaria 482-484, 491-497). Il commercio internazionale e la bilancia dei pagamenti internazionale 660-670.

3) Controllo dei costi in imprese industriali. Fanello, L'organizzazione di una piccola industria, Angeli (tutto il volume)

Costi: di produzione, dei prodotti, delle materie prime, del magazzino, della manodopera, delle macchine. Spese generali. Il calcolo dei costi.

Interdipendenza degli elementi di costo. Il controllo dei costi. I

prodotti intermedi. Il magazzino. Controllo del costo della manodopera. In particolare il controllo del costo di fabbricazione. Diagrammi di controllo.

La produzione su commessa. Punto di rottura fra prezzi di vendita e costi.

La riduzione dei costi. La produzione presso terzi. Gli appalti. La sostituzione di macchine a uomini. La programmazione di una impresa.

Macchinari. Spese di esercizio e spese di impianto. L'analisi della

produzione ed in particolare l'analisi della commessa di lavorazione. Scopo

della programmazione della produzione. La programmazione lunga, media e breve. La programmazione giornaliera. Un tipico organigramma.

## Bibliografia

Samuelson 1996, Economia, McGraw Hill (parti)

Fanello 1990, L'organizzazione di una piccola industria, Angeli.

Il **bilancio** in **forma abbreviata**, **Caso** e Gioffrè -  
ed. **Buffetti**

## Modalità di esame

orale -su appuntamento

---

## Economia dei sistemi industriali

Codice: PP326   Crediti: 6   Semestre: 1   Sigla: ESI

Settore disciplinare:            SECS-P/07 - Economia Aziendale

### Docente

Dionisia Cazzaniga Francesetti  
*dcf@di.unipi.it*  
*stanza n.394, dip.to di Informatica*  
*tel:0502212711*

### Prerequisiti

nozioni analisi

### Obiettivi di apprendimento

Creare un tecnico informatico facilmente inseribile nelle imprese manifatturiere o di servizio. Il soggetto deve essere in grado di analizzare un'impresa sulla base di cognizioni micro e macro economiche di base compreso il bilancio. Deve conoscere la formazione dei costi ed il loro controllo , nonchè come si organizza una impresa manifatturiera.

### Programma

1)Si premettono le nozioni sul bilancio;libro di testo: Caso e Giofrè: BILANCIO IN FORMA ABBREVIATA ed Buffetti

2) Nozioni di base di micro e macro economia. Testo: Samuelson, XV edizione 1996–sono indicate di seguito le pagine ed i capitoli da seguire.

Domanda e offerta. pg 40-52. Elasticità della domanda rispetto al prezzo 6, pg. 0-68. Domanda e comportamento del consumatore: dall'utilità al mercato, pg. 78-87. Curva indifferenza pg.92-95 Produzione e organizzazione delle imprese 101-109 Analisi economica dei costi 114-123 Comportamento di offerta dell'impresa concorrenziale 137-143 Concorrenza imperfetta e problema del monopolio 160-172 Calcolo del valore di un investimentooo valore attuale delle attività patrimoniali 258-263 Quadro generale di macroeconomia 390-401 Il prodotto interno lordo, criterio di misura 410-415 Il consumo e l'investimento 429-443 Elementi che incidono sulla domanda aggregata e loro rappresentazione nel moltiplicatore 452-462 (Moneta ed attività bancaria 482-484, 491-497). Il commercio internazionale e la bilancia dei pagamenti internazionale 660-670.

3) Controllo dei costi in imprese industriali. Fanello, L'organizzazione di una piccola industria, Angeli (tutto il volume)

Costi: di produzione, dei prodotti, delle materie prime, del magazzino, della manodopera, delle macchine. Spese generali. Il calcolo dei costi.

Interdipendenza degli elementi di costo. Il controllo dei costi. I

prodotti intermedi. Il magazzino. Controllo del costo della manodopera. In

particolare il controllo del costo di fabbricazione. Diagrammi di

controllo.

La produzione su commessa. Punto di rottura fra prezzi di vendita e costi.

La riduzione dei costi. La produzione presso terzi. Gli appalti. La sostituzione di macchine a uomini. La programmazione di una impresa.

Macchinari. Spese di esercizio e spese di impianto. L'analisi della

produzione ed in particolare l'analisi della commessa di lavorazione. Scopo

della programmazione della produzione. La programmazione lunga, media e

breve. La programmazione giornaliera. Un tipico organigramma.

## Bibliografia

Samuelson 1996, Economia, McGraw Hill (parti)

Fanello 1990, L'organizzazione di una piccola industria, Angeli.

Il bilancio in forma abbreviata, Caso e Giofrè -  
ed. Buffetti

## Modalità di esame

orale -su appuntamento

---

## Fisica A

Codice: BB001 Crediti: 6 Semestre: 2 Sigla: FIS

Settore disciplinare: FIS/02 - Fisica Teorica,  
Modelli e Metodi Matematici

## Docente

Paolo Rossi

*paolo.rossi@df.unipi.it*

*www.df.unipi.it/~rossi*

*Tel. 050 2214884/050 2214643*

## Prerequisiti

Conoscenze di base di Analisi Matematica, Algebra e Geometria

## Obiettivi di apprendimento

Introduzione ai principi e ai metodi della fisica classica

*Conoscenze.* Conoscenza delle leggi fondamentali della meccanica classica: cinematica e dinamica del punto materiale, legge di gravitazione universale. Conoscenza delle leggi fondamentali dell'elettromagnetismo classico: equazioni di Maxwell

*Capacità.* Capacità di impostare e risolvere, analiticamente e numericamente, semplici problemi di meccanica ed elettromagnetismo

*Comportamenti.* Lo studente dovrebbe acquisire un comportamento scientifico nell'individuazione dei problemi e nell'adozione di strategie volte alla loro analisi e alla loro soluzione

## Descrizione

Il corso è costituito da lezioni cattedratiche e da esercitazioni. Nelle lezioni vengono enunciati e spiegati i principi e vengono ricavate le leggi della meccanica e dell'elettromagnetismo. Nelle esercitazioni vengono illustrati con esempi pratici i metodi che permettono la risoluzione di problemi attinenti alla materia del corso. Vengono introdotti numerosi argomenti di natura matematica, in particolare di geometria analitica, di calcolo vettoriale e di analisi, che sono propedeutici all'introduzione del formalismo fisico.

## Indicazioni metodologiche

Valutare in corso di erogazione il livello di raggiungimento degli obiettivi utilizzando gli strumenti delle verifiche intermedie

## Programma

1. Introduzione Metodo sperimentale. L'operazione di misura. Unità e dimensioni delle osservabili. Descrizione matematica delle grandezze fisiche: sistemi di coordinate, vettori, prodotti scalari e vettoriali, derivate.
2. Cinematica del punto. Legge oraria, velocità, accelerazione. Moto rettilineo uniforme. Moto uniformemente accelerato. Moto in due dimensioni e moto di un proiettile. Moto circolare uniforme. Accelerazione tangenziale e radiale.
3. Dinamica del punto. Massa e quantità di moto. Conservazione della quantità di moto. Forze e secondo principio della dinamica. Legge di azione e reazione.
4. Applicazioni delle leggi della meccanica. Piano inclinato e reazioni vincolari. Moto armonico. Pendolo semplice. Forze di attrito.
5. Conservazione dell'energia. Lavoro ed energia cinetica. Il teorema delle forze vive. Applicazioni. Problemi d'urto elastico ed inelastico. Dipendenza del lavoro dal percorso. Forze conservative. Energia potenziale. Conservazione dell'energia meccanica. Alcune forze conservative e l'energia potenziale ad esse associata.
6. Gravitazione universale. Leggi di Keplero. Legge di Newton. Potenziale gravitazionale. Velocità di fuga.
7. Teoria cinetica dei gas e termodinamica. Legge dei gas perfetti. Concetto

- di temperatura e funzione di stato. Conservazione dell'energia e primo principio della termodinamica.
8. Elettrostatica. Cariche elettriche e legge di Coulomb. Definizione del campo elettrico. La legge di Gauss per i campi elettrici. Applicazioni. Moto di cariche in un campo elettrico e conservazione dell'energia. Energia elettrostatica. Condensatori. Capacità. Condensatori in serie e in parallelo.
  9. Corrente elettrica e resistenza. Corrente elettrica. Conservazione della corrente. Resistenza elettrica. Legge di Ohm ed effetto Joule. Resistenze in serie e in parallelo.
  10. Campo magnetico. Definizione del campo magnetico e forza di Lorentz. Proprietà della forza di Lorentz e conservazione dell'energia. Forza esercitata da un campo magnetico su una corrente elettrica. Campo magnetico generato da correnti. La forza tra due correnti parallele. Assenza di cariche magnetiche isolate. Legge di Ampere e teorema della circuitazione.
  11. Elettrodinamica. Induzione elettromagnetica. Legge di Faraday-Lenz.

Ore lezione: 30

Ore esercitazione: 15

Ore laboratorio: 0

Ore  
seminari: 0

### Bibliografia

Le dispense del corso redatte dal prof. Beverini sono disponibili nel sito <http://www.df.unipi.it/~beverini/SP/> che contiene anche testi e soluzioni degli esercizi proposti nelle prove d'esame

### Modalità di esame

Scritto e orale. L'esame può essere superato con il superamento delle due prove in itinere, e nella maggior parte dei casi il superamento della prova scritta (o delle prove in itinere) permette l'esenzione dalla prova orale. Le prove scritte sono costituite da cinque o sei problemi di fisica, per ognuno dei quali si deve rispondere a due o tre domande a risposta (numerica) multipla

Ulteriore pagina web del corso: <http://www.df.unipi.it/~rossi>

---

## Fisica B

Codice: BB001    Crediti: 6    Semestre: 2    Sigla: FIS

Settore disciplinare: FIS/02 - Fisica Teorica,  
Modelli e Metodi Matematici

### Docente

Giorgio Carelli  
[giorgio.carelli@df.unipi.it](mailto:giorgio.carelli@df.unipi.it)  
[www.df.unipi.it/~carelli/](http://www.df.unipi.it/~carelli/)  
Tel. 050-2214285

## Prerequisiti

Conoscenze matematiche di base relative al calcolo numerico, alla geometria euclidea ed alla trigonometria. Conoscenza del concetto generale di funzione matematica, derivata ed integrale. Calcolo delle funzioni elementari delle loro derivate ed integrali.

## Obiettivi di apprendimento

Introduzione al linguaggio della fisica come descrizione matematica dei fenomeni naturali. Conoscenza delle leggi fondamentali della meccanica, dell'elettricità e del magnetismo con particolare attenzione all'uso del concetto di conservazione delle grandezze fisiche.

*Conoscenze.* Conoscenze di base.

*Capacità.* Capacità di apprendimento, di comprensione e di verifica di un modello matematico della realtà.

*Comportamenti.*

## Descrizione

Il corso fornisce una introduzione sistematica ai principi fondanti della fisica classica. La verifica della comprensione dei concetti si basa su semplici applicazioni che richiedono conoscenze di algebra, geometria ed occasionalmente di analisi matematica. Gli argomenti affrontati sono: le leggi del moto, derivate dai principi di Newton, i concetti di lavoro ed energia, principi di conservazione di energia e quantità di moto; le interazioni tra cariche elettriche, la legge di Gauss ed il potenziale elettrico con particolare enfasi sul principio di sovrapposizione e sul concetto di simmetria; la risoluzione di semplici circuiti elettrici in continua; il campo magnetico. I fenomeni di induzione e le leggi che li descrivono.

## Indicazioni metodologiche

Il processo di apprendimento è organizzato in sequenza logica con valutazione intermedia degli obiettivi di apprendimento raggiunti.

## Programma

- Richiami di geometria analitica. Sistemi di riferimento cartesiani in due e tre dimensioni. Equazioni della retta, della circonferenza, dell'iperbole equilatera, della parabola. Tangente ad una curva. Rappresentazione parametrica.
- Grandezze fisiche e unità di misura. Misura di una grandezza. Grandezze fondamentali e grandezze derivate. Dimensioni fisiche di una grandezza. Sistemi coerenti di unità di misura. Il Sistema Internazionale.
- Cinematica. Descrizione del moto unidimensionale di un punto materiale. Legge oraria del moto. Definizione di velocità media e velocità istantanea. Definizione di accelerazione media e accelerazione istantanea. Il moto uniformemente accelerato. Moto nello spazio tridimensionale. Grandezze

- vettoriali e vettori. Il vettore spostamento. Somma e differenza di vettori. Velocità e accelerazione vettoriali. Moto circolare ed accelerazione centripeta. Moto uniformemente accelerato in 3 dimensioni.
- Dinamica del punto materiale. Primo e secondo principio della dinamica. Definizione e unità di misura delle forze. Moto di un grave. Forze vincolari e piano inclinato. Il terzo principio della dinamica. Tensione di una fune. Esempi di forze: forza gravitazionale, forza elettrostatica, forze elastiche. Moto armonico. Definizione di lavoro e di potenza e loro unità di misura SI. Lavoro della forza peso e della forza elastica. Teorema dell'energia cinetica. Forze posizionali. Concetto di campo di forze. Definizione di forze conservative. Definizione di energia potenziale. Energia potenziale gravitazionale e energia potenziale elastica. Principio di conservazione dell'energia meccanica. Definizione di impulso e di quantità di moto. Teorema dell'impulso. Urti elastici ed anelastici.
  - Corpi estesi. Sistemi di punti materiali. Centro di massa. Teorema del baricentro. Quantità di moto totale di un sistema. Principio di conservazione della quantità di moto. Corpo rigido. Momento di una forza. Statica di un corpo rigido.
  - Dinamica celeste. Le leggi di Keplero. Legge di gravitazione universale. Il momento della quantità di moto. Conservazione della quantità di moto nel moto di un satellite o di un pianeta. Energia potenziale del campo gravitazionale. Velocità di fuga.
  - Fluidi. Fluidi. Densità e pressione. Legge di Stevino e principio d'Archimede. Descrizione macroscopica e microscopica dei fenomeni. Modello cinetico dei gas. Temperatura assoluta. Legge dei gas perfetti.
  - Elettrostatica. I fenomeni elettrici. Corpi conduttori e isolanti Forza di Coulomb. Campo elettrico. Linee di forza. Teorema di Gauss e sue applicazioni ai corpi conduttori. Campo generato da una sfera conduttrice carica; da una sfera dielettrica uniform. carica; da una lastra conduttrice piana. Potenziale elettrostatico. Potenziale generato da una carica puntiforme e da un insieme di cariche puntiformi. Potenziale di un conduttore. Condensatori. Campo elettrico in un condensatore piano. Energia immagazzinata in un condensatore.
  - Corrente elettrica. Legge di Ohm. Potenza dissipata dalla corrente. Forza elettromotrice. Resistenze in serie e in parallelo. Carica di un condensatore
  - Elettromagnetismo. Forza di Lorentz. Forza magnetica agente su un elemento di corrente. Campo generato da un elemento di corrente. Legge di Biot-Savart. Forza tra due fili rettilinei percorsi da corrente. Legge di Ampère. Campo di un solenoide. Definizione di flusso del campo magnetico concatenato con un circuito. Induzione elettromagnetica: legge di Faraday e legge di Lenz. Spira rotante in campo magnetico.

## Bibliografia

Serway, "Principi di Fisica", EdiSES  
Halliday, "Principi di Fisica", Ambrosiana Editrice  
E. Amaldi, "Fisica Generale", Zanichelli

## Modalità di esame

Scritto e orale. L'esame può essere superato con il superamento delle due prove in itinere.

Ulteriore pagina web del corso:

<http://www.df.unipi.it/~carelli/didattica/inf1.html>

---

## Fondamenti di programmazione A

Codice: AA002   Crediti: 9   Semestre: 1   Sigla: FP

Settore disciplinare:            INF/01 - Informatica

### Docente

Roberto Barbuti  
*barbuti@di.unipi.it*  
*www.di.unipi.it/~barbuti*  
*stanza n. 334, dip.to di Informatica*  
*tel: 0502212747*

### Prerequisiti

nessuno

### Obiettivi di apprendimento

Il corso intende presentare i fondamenti della programmazione. A questo scopo sono illustrate le tecniche formali per descrivere le caratteristiche dei linguaggi di programmazione: grammatiche e automi per descrivere la sintassi e sistemi di transizione per descrivere la semantica. Le tecniche sono esemplificate su un sottoinsieme di Java. Sono poi introdotte, tramite esempi di strutture dati e di algoritmi, alcune tecniche fondamentali di programmazione iterativa e ricorsiva.

### Descrizione

- Algoritmi e problem-solving
- Elementi di sintassi e semantica dei linguaggi di programmazione
- Teoria degli automi
- Costrutti di base della programmazione
- Strutture di dati e gestione della memoria
- Ricorsione e strategie di calcolo basate sulla ricorsione

### Programma

- Introduzione
- Sintassi
- Automi
- Grammatiche { materiale didattico: D.Pedreschi, Elementi di Sintassi dei

- Linguaggi di Programmazione, Dispensa}
- Semantica Operazionale
  - Sistemi di transizione formalizzazione del nucleo iterativo di Java modello dello stato con stack e heap metodi e passaggio dei parametri {materiale didattico: R.Barbuti et. al Semantica Operazionale Dispensa}
  - Tipi di dato e problem solving: prima parte  
array  
metodi iterativi di search e sort  
tabelle hash
  - Ricorsione
  - Tipi di dato e problem solving: seconda parte metodi ricorsivi di sort  
liste e alberi binari  
alberi binari di ricerca {materiale didattico: C.S. Horstmann: Concetti di Informatica e fondamenti di Java 2, Apogeo A.Brogi et al. Introduzione alle strutture dati in Java, dispense}

### Modalità di esame

Scritto e orale

---

## Fondamenti di programmazione B

Codice: AA002   Crediti: 9   Semestre: 1   Sigla: FP

Settore disciplinare:      INF/01 - Informatica

### Docente

Paolo Mancarella  
*paolo@di.unipi.it*  
*www.di.unipi.it/~paolo*  
*stanza n.367, dip.to di Informatica*  
*tel:0502212710*

### Prerequisiti

nessuno

### Obiettivi di apprendimento

Il corso intende presentare i fondamenti della programmazione. A questo scopo sono illustrate le tecniche formali per descrivere le caratteristiche dei linguaggi di programmazione: grammatiche e automi per descrivere la sintassi e sistemi di transizione per descrivere la semantica. Le tecniche sono esemplificate su un sottoinsieme di Java. Sono poi introdotte, tramite esempi di strutture dati e di algoritmi, alcune tecniche fondamentali di programmazione iterativa e ricorsiva.

## Descrizione

- Algoritmi e problem-solving
- Elementi di sintassi e semantica dei linguaggi di programmazione
- Teoria degli automi
- Costrutti di base della programmazione
- Strutture di dati e gestione della memoria
- Ricorsione e strategie di calcolo basate sulla ricorsione

## Programma

- Introduzione
- Sintassi
- Automi
- Grammatiche {materiale didattico: D.Pedreschi, Elementi di Sintassi dei Linguaggi di Programmazione, Dispensa}
- Semantica Operazionale
- Sistemi di transizione formalizzazione del nucleo iterativo di Java modello dello stato con stack e heap metodi e passaggio dei parametri {materiale didattico: R.Barbuti et. al Semantica Operazionale Dispensa}
- Tipi di dato e problem solving: prima parte  
array  
metodi iterativi di search e sort  
tabelle hash
- Ricorsione
- Tipi di dato e problem solving: seconda parte metodi ricorsivi di sort  
liste e alberi binari  
alberi binari di ricerca {materiale didattico: C.S. Horstmann: Concetti di Informatica e fondamenti di Java 2, Apogeo A.Brogi et al. Introduzione alle strutture dati in Java, dispense}

## Bibliografia

Dispense fornite dai docenti scaricabili

alla pagina

<http://www.di.unipi.it/~paolo/FP/>

## Modalità di esame

Scritto e orale

Ulteriore pagina web del corso: <http://www.di.unipi.it/~paolo/didattica.html>

Codice: AA002 Crediti: 9 Semestre: 1 Sigla: FP

Settore disciplinare: INF/01 - Informatica

### Docente

Franco Turini  
*turini@di.unipi.it*  
*www.di.unipi.it/~turini*  
*stanza n.335, dip.to di Informatica*  
*tel:0502212753*

### Prerequisiti

nessuno

### Obiettivi di apprendimento

Il corso intende presentare i fondamenti della programmazione. A questo scopo sono illustrate le tecniche formali per descrivere le caratteristiche dei linguaggi di programmazione: grammatiche e automi per descrivere la sintassi e sistemi di transizione per descrivere la semantica. Le tecniche sono esemplificate su un sottoinsieme di Java. Sono poi introdotte, tramite esempi di strutture dati e di algoritmi, alcune tecniche fondamentali di programmazione iterativa e ricorsiva.

### Descrizione

- Algoritmi e problem-solving
- Elementi di sintassi e semantica dei linguaggi di programmazione
- Teoria degli automi
- Costrutti di base della programmazione
- Strutture di dati e gestione della memoria
- Ricorsione e strategie di calcolo basate sulla ricorsione

### Programma

- Introduzione
- Sintassi
- Automi
- Grammatiche {materiale didattico: D.Pedreschi, Elementi di Sintassi dei Linguaggi di Programmazione, Dispensa}
- Semantica Operazionale
- Sistemi di transizione formalizzazione del nucleo iterativo di Java modello dello stato con stack e heap metodi e passaggio dei parametri {materiale didattico: R.Barbuti et. al Semantica Operazionale Dispensa}
- Tipi di dato e problem solving: prima parte  
array  
metodi iterativi di search e sort  
tabelle hash
- Ricorsione
- Tipi di dato e problem solving: seconda parte metodi ricorsivi di sort



## Informatica e cultura

Codice: AA488   Crediti: 3   Semestre: 1   Sigla: ICU

Settore disciplinare:            INF/01 - Informatica

### Docente

Francesco Romani  
*romani@di.unipi.it*  
*www.di.unipi.it/~romani*  
*stanza n.326, dip.to di Informatica*  
*tel: 0502212734*

### Prerequisiti

nessuno

### Obiettivi di apprendimento

E' un dato di fatto: le tecnologie avanzate di oggi saranno le tecnologie superate di domani. Bisogna sfatare la convinzione che l'informatica sia solo super tecnologia: un meccanismo perverso che crea meravigliose-misteriose macchine che ci cambiano (spesso in peggio) la vita. Questo corso si rivolge sia agli studenti di materie scientifiche che a quelli di materie umanistiche e ha lo scopo di mostrare come le radici concettuali dell'informatica siano indissolubilmente legate alle radici della nostra civiltà (linguaggio, immagine, scrittura ... filosofia, logica, matematica). L'obiettivo non è fornire una visione coerente e completa di argomenti tanto vasti e complessi da abbracciare gran parte dello scibile umano, quanto di mostrare l'esistenza di connessioni e problematiche che spesso sfuggono, con la speranza di stimolare il desiderio di approfondimenti personali.

*Conoscenze.* Una visione di insieme delle relazione tra l'informatica e le altre discipline umanistiche e scientifiche.

*Capacità.* Il corso dovrebbe stimolare la capacità di approfondimento personale su argomenti di tipo generale.

*Comportamenti.* L'ampliamento delle conoscenze dei fondamenti filosofici dell'informatica non dovrebbe avere effetti sui comportamenti degli allievi.

### Descrizione

Il corso si articola in 10 seminari di circa 60 minuti seguiti da una discussione. Le presentazioni sono "multimediali" con inserti musicali, immagini e videoclip. Abbondano anche le citazioni dei classici (Saffo, Sofocle, Platone, Dante, Galileo, Shakespeare, Cartesio ...)

## Indicazioni metodologiche

Il processo di apprendimento è organizzato in moduli posti in sequenza logica.

## Programma

- Le radici dell'informazione e del suo trattamento
  - Il sultano e le lumache
  - La cosa più bella ...
  - ...oscura tutto il resto
  - L'informazione è leggera ...
  - ...ma non priva di massa
  - Il tutto è maggiore della somma delle sue parti?
  - Ciò che è possibile è obbligatorio
  - *The rise of the machines*
- Le prime *tecnologie*: parole e immagini
  - Realtà e rappresentazione
  - Lessico, sintassi, semantica
  - Le immagini
  - Tecnologia dell'immagine attraverso i tempi
  - Il fondamento di ogni rappresentazione
- L'evoluzione della scrittura e della codifica
  - L'invenzione della scrittura
  - Ideogrammi e lettere: dall'immagine al simbolo
  - Linguaggio e scrittura
  - Alfabeti - Stringhe - Bit - Byte
  - Il concetto di codifica
  - Goedelizzazione
- Il concetto di numero, la sua evoluzione nella civiltà
  - Uno, due ... pochi, tanti
  - Sassi e pecore
  - I numeri cardinali: corrispondenza biunivoca
  - I numeri ordinali: uno dopo l'altro
  - La matematica è una costruzione o una scoperta?
  - Numeri naturali ed estensioni
  - Visione costruttiva ed esistenziale
  - Paradossi numerici
  - Infinito e infiniti
- Il "calcolo" dall'algoritmo di Euclide al software
  - Il concetto di algoritmo
  - Algoritmi antichi
  - Macchine e programmi
  - Tesi di Church-Turing
  - I teoremi di Goedel
- Le difficoltà intrinseche del fare calcolo
  - Calcolabilità e decidibilità
  - Complessità in tempo
  - Complessità in spazio
  - Complessità *program size*
- Intelligenza naturale e artificiale: una nuova visione del test di Turing
  - Medium cognitivo

- Fenomenologia degli oggetti non osservati
- Approccio dogmatico al pensiero
- Approccio comportamentista al pensiero
- Test di Turing implicito
- Test di Turing inverso
- Mente e linguaggio - Test di Turing linguistico
- Rigiriamo la frittata! - Test di Turing assoluto
- Il caso nell'informatica e nella realtà
  - Casualità, Causalità e Pseudocasualità
  - Il Lotto (*la madre dei fessi è sempre incinta*)
  - Entropia e bit
  - Rumore e informazione
  - Il Caos deterministico
  - Entropia ed evoluzione
  - La freccia del tempo
- Vantaggi e problematiche dell'uso del computer nella didattica
  - E - ducare: opinioni a confronto
  - Computer e apprendimento programmato
  - Computer e apprendimento per scoperta
  - In medio stat virtus
  - Macchine volanti e macchine per imparare
  - Ruote per la mente

Ore lezione: 20

Ore esercitazione: 0

Ore laboratorio: 0

Ore  
seminari: 0

## Bibliografia

**Bibliografia** *La bibliografia si intende come spunto di ricerca, ovviamente non è necessario consultarla tutta per superare l'esame, alcuni testi sono fuori catalogo e di difficile reperimento*

- **L. Brillouin**, Science and Information Theory, [Academic Press](#)
- **E. Boerger**, Computability, Complexity, Logic. Studies in Logic and the Foundations of Mathematics, [North-Holland](#)
- **F. Celi, F. Romani**, Macchine per imparare, [Erickson](#)
- **R. Dawkins**, Il gene egoista, [Mondadori](#)
- **P. Ferragina, F. Luccio**, Crittografia, [Bollati Boringhieri](#)
- **S. R. Fischer**, Breve storia del linguaggio, [Utet](#)
- **L. Geymonat**, Storia del pensiero filosofico e scientifico, [Garzanti](#)
- **G. C. Ghirardi**, Un'occhiata alle carte di Dio, [Il Saggiatore](#)
- **D. R. Hofstadter**, D. C. Dennet, L'Io della mente, [Adelphi](#)
- **D. R. Hofstadter**, Goedel, Escher, Bach, [Adelphi](#)
- **F. Jacob**, La logica del vivente, [Einaudi](#)
- **M. R. Laganà, M. Righi, F. Romani**, Informatica: concetti e sperimentazioni, [Apogeo](#)
- **G. O. Longo**, Il nuovo Golem, [Laterza](#)
- **J. Monod**, Il caso e la necessità, [Mondadori](#)
- **G. Toraldo di Francia**, Le cose e i loro nomi, [Laterza](#)
- **P. Zellini**, Breve storia dell'infinito, [Adelphi](#)
- **P. Zellini**, La ribellione del numero, [Adelphi](#)

## Modalità di esame

Per superare l'esame è necessario presentare una breve relazione critica o di approfondimento (15 - 30 pagine, in buon italiano) su un argomento a scelta tra quelli presentati o affine ad essi.

Ulteriore pagina web del corso: <http://www.di.unipi.it/~romani/ICU/ICU.html>

---

## Ingegneria del software A

Codice: AA017   Crediti: 6   Semestre: 1   Sigla: Is

Settore disciplinare:            INF/01 - Informatica

### Docenti

Laura Semini  
*semini@di.unipi.it*  
*www.di.unipi.it/~semini*  
*stanza n. 329a, dip.to di Informatica*  
*tel: 0502212783*

Vincenzo Ambriola  
*ambriola@di.unipi.it*  
*www.di.unipi.it/~ambriola*  
*stanza n. 320, dip.to di Informatica*  
*tel: 0502212751*

### Prerequisiti

Metodologie di programmazione

### Obiettivi di apprendimento

Il corso presenta un'introduzione completa alle tematiche dell'ingegneria del software fornendo le basi per comprendere e partecipare a un moderno processo di sviluppo software e realizzare prodotti software che soddisfino requisiti di qualità predefiniti.

*Conoscenze.* Lo studente acquisirà conoscenze sui principali modelli di sviluppo software.

*Capacità.* Lo studente saprà utilizzare notazioni di modellazione per l'analisi e la progettazione, come UML 2.

*Comportamenti.* Lo studente saprà essere indipendente nella valutazione delle scelte di progetto.

### Descrizione

Il corso (formato da due moduli) presenta un'introduzione completa alle diverse tematiche dell'ingegneria del software fornendo le basi per

comprendere e partecipare a un moderno processo di sviluppo software e realizzare prodotti software che soddisfino requisiti di qualità predefiniti.

## Programma

Il corso è organizzato in due moduli: il primo tenuto dalla prof. Laura Semini, il secondo dal prof. Ambriola.

- *Unified modelling language (UML 2)*
- *Progettazione*: Aspetti generali della progettazione. Progettazione orientata agli oggetti.
- *Verifiche e prove*: Obiettivi e pianificazione delle verifiche. Ispezione del codice. Progettazione e valutazione delle prove.
- *Controllo della qualità*: Misurazione del software. Modelli per la stima. Qualità dei prodotti e dei processi software.

## Bibliografia

Capitoli scelti dai seguenti testi:

- *R.S. Pressman, Principi di Ingegneria del software (quarta edizione italiana), McGraw-Hill, 2004.*
- *M. Fowler, UML Distilled (terza edizione italiana), Pearson, 2004.*
- *D. Rosemberg, K. Scott, Use Case Driven Object Modeling with UML, Addison Wesley, 1999.*
- *E. Gamma, R. Helm, R. Johnson, J. Vlissides. Design Patterns (prima edizione italiana). Addison Wesley, 2002.*

Dispense a cura dei docenti:

- *L. Semini, [Architetture software](#). Nota didattica, 2005.*
- *V. Ambriola, C. Montangero, L. Semini, [Esercizi di Ingegneria del Software. Parte I](#). Nota didattica, 2005.*
- *V. Ambriola, C. Montangero, L. Semini, [Esercizi di Ingegneria del Software. Parte II](#). Nota didattica, 2005.*

## Modalità di esame

La valutazione avviene mediante prova scritta e prova orale. La prova scritta può essere sostituita dal superamento di due verifiche in itinere.

Ulteriore pagina web del corso: <http://www.di.unipi.it/~ambriola/is/radice.htm>

---

## Ingegneria del software B

Codice: AA017   Crediti: 6   Semestre: 1   Sigla: Is

### Docente

Carlo Montangero  
*monta@di.unipi.it*  
*www.di.unipi.it/~monta*  
stanza n.293, dip.to di Informatica  
tel: 0502212745

### Prerequisiti

Metodologie di programmazione.

### Obiettivi di apprendimento

Il corso presenta un'introduzione completa alle tematiche dell'ingegneria del software fornendo le basi per comprendere e partecipare a un moderno processo di sviluppo software e realizzare prodotti software che soddisfino requisiti di qualità predefiniti.

*Conoscenze.* Lo studente acquisirà conoscenze sui principali modelli di sviluppo software.

*Capacità.* Lo studente saprà utilizzare notazioni di modellazione per l'analisi e la progettazione, come UML 2.

*Comportamenti.* Lo studente saprà essere indipendente nella valutazione delle scelte di progetto.

### Programma

- *Processo di sviluppo software:* Problemi della produzione del software. Modelli di ciclo di vita.
- *Unified modelling language (UML 2)*
- *Analisi e progettazione:* Aspetti generali dell'analisi e della progettazione. Analisi e progettazione orientata agli oggetti.
- *Verifiche e prove:* Obiettivi e pianificazione delle verifiche. Ispezione del codice. Progettazione e valutazione delle prove.
- *Controllo della qualità:* Misurazione del software. Modelli per la stima. Qualità dei prodotti e dei processi software.

Ore lezione: 34 Ore esercitazione: 14

### Bibliografia

Capitoli scelti dai seguenti testi:

- *R.S. Pressman, Principi di Ingegneria del software (quarta edizione italiana), McGraw-Hill, 2004.*
- *M. Fowler, UML Distilled (terza edizione italiana), Pearson, 2004.*
- *D. Rosemberg, K. Scott, Use Case Driven Object Modeling with UML, Addison Wesley, 1999.*

- *E. Gamma, R. Helm, R. Johnson, J. Vlissides. Design Patterns (prima edizione italiana). Addison Wesley, 2002.*

Dispense a cura dei docenti:

- *L. Semini, [Architetture software](#). Nota didattica, 2005.*
- *V. Ambriola, C. Montangero, L. Semini, [Esercizi di Ingegneria del Software. Parte I](#). Nota didattica, 2005.*
- *V. Ambriola, C. Montangero, L. Semini, [Esercizi di Ingegneria del Software. Parte II](#). Nota didattica, 2005.*

### Modalità di esame

La valutazione avviene mediante prova scritta e prova orale. La prova scritta può essere sostituita dal superamento di due verifiche in itinere.

Ulteriore pagina web del corso:

[http://www.di.unipi.it/~monta/IS2005/IS\\_files/HOME.htm](http://www.di.unipi.it/~monta/IS2005/IS_files/HOME.htm)

---

## Introduzione all'audio digitale

Codice: AA021   Crediti: 3   Semestre: 2   Sigla: IAD

Settore disciplinare:      INF/01 - Informatica

### Docente

Francesco Romani  
*romani@di.unipi.it*  
*www.di.unipi.it/~romani*  
*stanza n. 326, dip. to di Informatica*  
*tel: 0502212734*

### Prerequisiti

Conoscenze base di Analisi Matematica

### Obiettivi di apprendimento

Fornire un'idea degli aspetti matematici nascosti dietro il trattamento dell'audio digitale. Il materiale è presentato in forma discorsiva omettendo le dimostrazioni ma non rinunciando al rigore degli enunciati. Viene privilegiata la presentazione grafica dei fenomeni.

*Conoscenze.* Avere un'idea degli aspetti matematici nascosti dietro il trattamento dell'audio digitale.

*Capacità.* Capacità di orientarsi criticamente tra le offerte del mercato in termini di apparecchiature audio.

*Comportamenti.* La conoscenza dei fondamenti dell'audio digitale non dovrebbe

avere effetti sui comportamenti degli allievi.

### Descrizione

Il corso tratta dei fondamenti dell'audio digitale, sia dal punto di vista teorico (dominio del tempo, dominio della frequenza, trasformate di Fourier, quantizzazione campionamento, conversioni A/D e D/A) sia dal punto di vista della applicazioni pratiche, professionali e amatoriali.

### Indicazioni metodologiche

Il processo di apprendimento è organizzato in moduli posti in sequenza logica.

### Programma

- Segnali finiti e infiniti, potenza e energia di un segnale;
- dominio del tempo e dominio della frequenza;
- campionamento, teorema del campionamento e sue limitazioni pratiche;
- quantizzazione, rumore di quantizzazione;
- conversione A/D e D/A;
- dithering e noise shaping, tecniche avanzate di conversione;
- formati audio lineari e compressi;
- trattamento di segnali audio con personal computer (a livello professionale, semi-professionale e didattico);

Ore lezione: 20

Ore esercitazione: 0

Ore laboratorio: 0

Ore  
seminari: 0

### Bibliografia

Materiale fornito dal docente

### Modalità di esame

Presentazione di una relazione scritta su un argomento attinente al corso

Ulteriore pagina web del corso:

<http://www.cli.di.unipi.it/~romani/IAD/IAD.html>

---

## Laboratorio di applicazioni internet

Codice: AA051   Crediti: 6   Semestre: 2   Sigla: ISI

Settore disciplinare:      INF/01 - Informatica

### Docente

Tito Flagella  
*tito@di.unipi.it*

## Obiettivi di apprendimento

Il corso introduce le principali architetture, protocolli e strumenti di sviluppo per la realizzazione di Applicazioni Internet. Permettere inoltre agli studenti di esercitarsi nell'uso degli stessi in una serie di esercitazioni assistite ed in un progetto didattico completo.

*Conoscenze.* Lo studente acquisirà conoscenze sui principali protocolli e linguaggi utilizzati su Internet (in particolare su http), e sulle modalità architetture con cui un'applicazione Internet vada progettata.

*Capacità.* Lo studente saprà utilizzare i vari middleware e API necessari per la realizzazione di un'applicazione Internet completa in ambiente java (servlet API, jsp, JDBC, etc.), oltre a gli strumenti per l'interoperabilità con applicazioni terze (CORBA e Web Services).

*Comportamenti.* Lo studente saprà operare nella progettazione e nel tracciamento diagnostico di eventuali problemi applicativi, indipendentemente dagli specifici wizard e ambienti di programmazione, che tipicamente nascondono molte delle problematiche di realizzazione e di integrazione, costringendo poi lo sviluppatore a dipendere dallo specifico ambiente utilizzato

## Descrizione

<http://isi-www.cli.di.unipi.it/slides/intro.pdf>

## Indicazioni metodologiche

Per conseguire gli obiettivi indicati nel tempo a disposizione, sarà necessario:

- realizzare lezioni non soltanto teoriche, ma anche condotte come analisi a "run time" di applicazioni reali, che evidenzino le problematiche di integrazione trattate di volta in volta.
- Realizzare esercitazioni in Laboratorio che permettano agli studenti di superare le difficoltà d'uso dei tanti nuovi strumenti analizzati.

## Programma

### Introduzione

- L'evoluzione delle architetture software: dai mainframe al client/server a multi-tier/Internet
- L'impatto dei Firewall sulla progettazione delle applicazioni Internet

### Primo Livello Applicativo (Browser)

Introduzione alle tecnologie di base:

- il protocollo http
- richiami dei linguaggi html, javascript e xml

- applet java

### **Secondo Livello Applicativo (Web Server)**

- Il Web server come base dell'Application Server
- Introduzione alle web server extension: CGI, ISAPI, NSAPI, ASP, Servlet, JSP, PHP.
- Approfondimenti sulla Servlet API
- Introduzione a Java Template Engine: uso di JSP come template engine
- Interazione Client/Server: comunicazione Applet/Servlet tramite serializzazione di oggetti java su http
- Session Management: Uso delle sessioni per il riconoscimento delle transazioni http

### **Terzo Livello Applicativo (Back-End)**

- Richiami sulla programmazione di transazioni su RDBMS
- L'uso di JDBC per l'accesso a Database da Servlet
- L'uso di connection pools nelle Applicazioni Internet

### **Ulteriori Livelli Applicativi (Business Objects e Web Services)**

- Introduzione agli ORB e alle problematiche di integrazione tra applicazioni diverse
- L'architettura CORBA/IDL
- L'architettura SOAP/WSDL

### **Le problematiche di Sicurezza su Internet**

Richiami ai concetti di base:

- Insicurezze di Internet
- L'impatto ambientale dei Firewall sulla progettazione delle applicazioni Internet
- Certificati Digitali e standard X/509
- Richiami sull'uso dei Protocolli SSL/TLS e HTTPS
  
- La programmazione SSL in Java: JSSE

Ore lezione: 24 Ore esercitazione: 24

#### **Bibliografia**

reperibile alla pagina web del corso

#### **Modalità di esame**

La valutazione avverrà tramite la valutazione di progetti svolti dagli studenti autonomamente o in gruppi di due.

Ulteriore pagina web del corso: <http://projects.cli.di.unipi.it/isi/>

---

## Laboratorio di introduzione alla programmazione (nuovo) A

Codice: AA487   Crediti: 6   Semestre: 1-2   Sigla: LIP

Settore disciplinare: INF/01 - Informatica

### Obiettivi di apprendimento

Il corso è mirato a preparare gli studenti all'uso di un linguaggio di programmazione orientato agli oggetti per la risoluzione di problemi mediante il calcolatore. Il linguaggio di programmazione utilizzato è Java, di cui verranno introdotti i concetti essenziali. L'ambiente di programmazione è basato sul sistema operativo Linux.

*Conoscenze.* Alla fine del corso gli studenti avranno acquisito le basi necessarie alla risoluzione di semplici problemi e piccoli progetti mediante la progettazione e scrittura di programmi ben strutturati, nonché alla loro compilazione, esecuzione e debugging.

*Capacità.* Alla fine del corso gli studenti avranno acquisito la capacità di gestire il proprio account su sistemi Windows e Linux, di progettare, comprendere e scrivere codice Java, di sviluppare applicazioni su ambienti integrati di sviluppo usati anche in ambito aziendale (Eclipse) sfruttandone al meglio le caratteristiche di scrittura assistita e debugging.

*Comportamenti.* Alla fine del corso gli studenti avranno acquisito la capacità di lavorare sia in maniera autonoma che coordinata alla risoluzione di problemi mediante il calcolatore. Saranno inoltre coscienti delle problematiche inerenti la scrittura di programmi corretti e efficienti.

### Indicazioni metodologiche

Il corso introduce le basi della programmazione orientata agli oggetti usando Java come linguaggio di riferimento (su piattaforma Linux).

Il corso è diviso in due moduli, che si svolgono rispettivamente nel primo e nel secondo semestre dell'anno accademico con quattro ore settimanali.

Le quattro ore sono ripartite in due ore di teoria e due di esercitazioni.

- Nelle *ore di teoria* il docente presenta gli argomenti del programma.
- Le *ore di esercitazione* si svolgono in una aula informatizzata dove gli studenti possono esercitarsi al calcolatore risolvendo individualmente esercizi di programmazione.

Ore lezione: 18   Ore esercitazione: 18   Ore laboratorio: 36

### Modalità di esame

Durante lo svolgimento del corso, sia nel primo che nel secondo modulo, alcune ore di esercitazione saranno usate per le prove in itinere che serviranno a valutare l'apprendimento degli studenti. Alla fine del corso, agli studenti che avranno raggiunto la sufficienza nelle prove in itinere verrà assegnato un progetto. Il corso prevede l'obbligo di frequenza, da cui sono esonerati solo gli studenti lavoratori.

Ulteriore pagina web del corso: <http://www.cli.di.unipi.it/~rbruni/LIP-A-2005/>

### **Moduli:**

#### **Laboratorio di introduzione alla programmazione - 1 modulo A**

##### **Docente**

Roberto Bruni  
*bruni@di.unipi.it*  
*www.di.unipi.it/~bruni*  
*stanza n.319, dip.to di Informatica*  
*tel: 0502212785*

##### **Programma**

#### **Argomenti del I modulo**

- Descrizione del Centro di Calcolo
- Breve introduzione all'uso di Linux [e di Windows] nel Centro di Calcolo
- Il file system: directory e file
- Lo shell: alcuni comandi
- Alcune applicazioni e utilities (**Emacs**, **Pine**)
- Breve introduzione alla programmazione
- Introduzione al linguaggio Java, e all'ambiente di sviluppo (**Eclipse**)
- Tipi di dati primitivi e stringhe
- Strutture di controllo:
- Comandi condizionali (**if** e **switch**)
- Comandi iterativi (**for**, **while** e **do**)
- Ricorsione

- Debugging e documentazione del codice in **Eclipse**
- Uso di array
- Metodi: overloading e passaggio di parametri
- Introduzione ad alcune classi standard di Java

Ulteriore pagina web del corso: <http://www.cli.di.unipi.it/~rbruni/LIP-A-2005/>

## Laboratorio di introduzione alla programmazione - 2 modulo A

### Docente

Roberto Bruni  
*bruni@di.unipi.it*  
*www.di.unipi.it/~bruni*  
*stanza n.319, dip.to di Informatica*  
*tel:0502212785*

### Programma

- Introduzione alla programmazione orientata agli oggetti
- Classi e oggetti
- Input/Output su file
- Ereditarietà
- Gestione delle eccezioni
- Introduzione ad alcune classi standard di Java
- Struttura e documentazione di un buon progetto

---

## Laboratorio di introduzione alla programmazione (nuovo) B

Codice: AA487   Crediti: 6   Semestre: 1-2   Sigla: LIP

Settore disciplinare:      INF/01 - Informatica

## Obiettivi di apprendimento

Il corso è mirato a preparare gli studenti all'uso di un linguaggio di programmazione orientato agli oggetti per la risoluzione di problemi mediante il calcolatore. Il linguaggio di programmazione utilizzato è Java, di cui verranno introdotti i concetti essenziali. L'ambiente di programmazione è basato sul sistema operativo Linux.

*Conoscenze.* Alla fine del corso gli studenti avranno acquisito le basi necessarie alla risoluzione di semplici problemi e piccoli progetti mediante la progettazione e scrittura di programmi ben strutturati, nonché alla loro compilazione, esecuzione e debugging.

*Capacità.* Alla fine del corso gli studenti avranno acquisito la capacità di gestire il proprio account su sistemi Windows e Linux, di progettare, comprendere e scrivere codice Java, di sviluppare applicazioni su ambienti integrati di sviluppo usati anche in ambito aziendale (Eclipse) sfruttandone al meglio le caratteristiche di scrittura assistita e debugging

*Comportamenti.* Alla fine del corso gli studenti avranno acquisito la capacità di lavorare sia in maniera autonoma che coordinata alla risoluzione di problemi mediante il calcolatore. Saranno inoltre coscienti delle problematiche inerenti la scrittura di programmi corretti e efficienti.

## Descrizione

Il corso introduce le basi della programmazione orientata agli oggetti usando Java come linguaggio di riferimento (su piattaforma Linux).

Il corso è diviso in due moduli, che si svolgono rispettivamente nel primo e nel secondo semestre dell'anno accademico 2004/05 con quattro ore settimanali.

Le quattro ore sono ripartite in due ore di teoria e due di esercitazioni.

- Nelle *ore di teoria* il docente presenta gli argomenti del programma.
- Le *ore di esercitazione* si svolgono in una aula informatizzata dove gli studenti possono esercitarsi al calcolatore risolvendo individualmente esercizi di programmazione.

Durante lo svolgimento del corso, sia nel primo che nel secondo modulo, alcune ore di esercitazione saranno usate per le *prove in itinere* che serviranno a valutare l'apprendimento degli studenti. Alla fine del corso, agli studenti che avranno raggiunto la sufficienza nelle prove in itinere verrà assegnato un progetto. Il corso prevede l'obbligo di frequenza, da cui sono esonerati solo gli studenti lavoratori.

Ore lezione: 18   Ore esercitazione: 18   Ore laboratorio: 36

## Modalità di esame

Durante lo svolgimento del corso, sia nel primo che nel secondo modulo, alcune ore di esercitazione saranno usate per le prove in itinere che serviranno a valutare l'apprendimento degli studenti. Alla fine del corso, agli studenti che

avranno raggiunto la sufficienza nelle prove in itinere verrà assegnato un progetto. Il corso prevede l'obbligo di frequenza, da cui sono esonerati solo gli studenti lavoratori.

Ulteriore pagina web del corso: <http://www.cli.di.unipi.it/~acorradi/LIP-B-05>

### **Moduli:**

#### **Laboratorio di introduzione alla programmazione - 1 modulo B**

##### **Docente**

Andrea Corradini  
*andrea@di.unipi.it*  
*www.di.unipi.it/~andrea*  
*stanza n.357, dip.to di Informatica*  
*tel:0502212786*

##### **Programma**

- Descrizione del Centro di Calcolo
- Breve introduzione all'uso di Linux [e di Windows] nel Centro di Calcolo
- Il file system: directory e file
- Lo shell: alcuni comandi
- Alcune applicazioni e utilities (**Emacs, Pine**)
- Breve introduzione alla programmazione
- Introduzione al linguaggio Java, e all'ambiente di sviluppo (**Eclipse**)
- Tipi di dati primitivi e stringhe
- Strutture di controllo:
- Comandi condizionali (**if** e **switch**)
- Comandi iterativi (**for, while** e **do**)
- Ricorsione
- Debugging e documentazione del codice in **Eclipse**
- Uso di array
- Metodi: overloading e passaggio di parametri

- Introduzione ad alcune classi standard di Java

Ulteriore pagina web del corso: <http://www.cli.di.unipi.it/~acorradi/LIP-04/>

## Laboratorio di introduzione alla programmazione - 2 modulo B

### Docente

Andrea Corradini  
*andrea@di.unipi.it*  
*www.di.unipi.it/~andrea*  
*stanza n. 357, dip.to di Informatica*  
*tel: 0502212786*

### Obiettivi di apprendimento

Il corso è mirato a preparare gli studenti all'uso di un linguaggio di programmazione orientato agli oggetti per la risoluzione di semplici problemi mediante il calcolatore. Alla fine del corso gli studenti avranno acquisito le basi necessarie alla scrittura di semplici programmi, nonché alla loro compilazione, esecuzione e debugging. Il linguaggio di programmazione utilizzato è Java, di cui verranno introdotti i concetti essenziali. L'ambiente di programmazione è basato sul sistema operativo Linux.

### Programma

- Introduzione alla programmazione orientata agli oggetti
- Classi e oggetti
- Input/Output su file
- Ereditarietà
- Gestione delle eccezioni
- Introduzione ad alcune classi standard di Java
- Struttura e documentazione di un buon progetto

Ulteriore pagina web del corso: <http://www.cli.di.unipi.it/~acorradi/LIP-04/>

---

# Laboratorio di introduzione alla programmazione (nuovo) C

Codice: AA487 Crediti: 6 Semestre: 1-2 Sigla: LIP

Settore disciplinare: INF/01 - Informatica

## Obiettivi di apprendimento

Il corso è mirato a preparare gli studenti all'uso di un linguaggio di programmazione orientato agli oggetti per la risoluzione di problemi mediante il calcolatore. Il linguaggio di programmazione utilizzato è Java, di cui verranno introdotti i concetti essenziali. L'ambiente di programmazione è basato sul sistema operativo Linux.

*Conoscenze.* Alla fine del corso gli studenti avranno acquisito le basi necessarie alla risoluzione di semplici problemi e piccoli progetti mediante la progettazione e scrittura di programmi ben strutturati, nonché alla loro compilazione, esecuzione e debugging.

*Capacità.* Alla fine del corso gli studenti avranno acquisito la capacità di gestire il proprio account su sistemi Windows e Linux, di progettare, comprendere e scrivere codice Java, di sviluppare applicazioni su ambienti integrati di sviluppo usati anche in ambito aziendale (Eclipse) sfruttandone al meglio le caratteristiche di scrittura assistita e debugging.

*Comportamenti.* Alla fine del corso gli studenti avranno acquisito la capacità di lavorare sia in maniera autonoma che coordinata alla risoluzione di problemi mediante il calcolatore. Saranno inoltre coscienti delle problematiche inerenti la scrittura di programmi corretti e efficienti.

## Descrizione

Il corso introduce le basi della programmazione orientata agli oggetti usando Java come linguaggio di riferimento (su piattaforma Linux).

Il corso è diviso in due moduli, che si svolgono rispettivamente nel primo e nel secondo semestre dell'anno accademico 2004/05 con quattro ore settimanali.

Le quattro ore sono ripartite in due ore di teoria e due di esercitazioni.

- Nelle *ore di teoria* il docente presenta gli argomenti del programma.
- Le *ore di esercitazione* si svolgono in una aula informatizzata dove gli studenti possono esercitarsi al calcolatore risolvendo individualmente esercizi di programmazione.

Durante lo svolgimento del corso, sia nel primo che nel secondo modulo, alcune ore di esercitazione saranno usate per le *prove in itinere* che serviranno a valutare l'apprendimento degli studenti. Alla fine del corso, agli studenti che avranno raggiunto la sufficienza nelle prove in itinere verrà assegnato un progetto. Il corso prevede l'obbligo di frequenza, da cui sono esonerati solo gli studenti lavoratori.

Ore lezione: 18    Ore esercitazione: 18    Ore laboratorio: 36

### Modalità di esame

Durante lo svolgimento del corso, sia nel primo che nel secondo modulo, alcune ore di esercitazione saranno usate per le prove in itinere che serviranno a valutare l'apprendimento degli studenti. Alla fine del corso, agli studenti che avranno raggiunto la sufficienza nelle prove in itinere verrà assegnato un progetto. Il corso prevede l'obbligo di frequenza, da cui sono esonerati solo gli studenti lavoratori.

Ulteriore pagina web del corso: <http://www.di.unipi.it/~zavanell/>

### *Moduli:*

#### Laboratorio di introduzione alla programmazione - 1 modulo C

##### Docente

Andrea Zavanella  
*zavanell@di.unipi.it*  
*www.di.unipi.it/~zavanell*  
*tel: 0502212700*

##### Programma

- Descrizione del Centro di Calcolo
- Breve introduzione all'uso di Linux [e di Windows] nel Centro di Calcolo
- Il file system: directory e file
- Lo shell: alcuni comandi
- Alcune applicazioni e utilities (**Emacs**, **Pine**)
- Breve introduzione alla programmazione
- Introduzione al linguaggio Java, e all'ambiente di sviluppo (**Eclipse**)
- Tipi di dati primitivi e stringhe
- Strutture di controllo:
- Comandi condizionali (**if** e **switch**)
- Comandi iterativi (**for**, **while** e **do**)
- Ricorsione

- Debugging e documentazione del codice in **Eclipse**
- Uso di array
- Metodi: overloading e passaggio di parametri
- Introduzione ad alcune classi standard di Java

## Laboratorio di introduzione alla programmazione - 2 modulo C

### Docente

Andrea Zavanella  
*zavanell@di.unipi.it*  
*www.di.unipi.it/~zavanell*  
*tel: 0502212700*

### Programma

- Introduzione alla programmazione orientata agli oggetti
- Classi e oggetti
- Input/Output su file
- Ereditarietà
- Gestione delle eccezioni
- Introduzione ad alcune classi standard di Java
- Struttura e documentazione di un buon progetto

---

## Laboratorio di linguaggi di sistema A

Codice: AA536   Crediti: 3   Semestre: 1   Sigla: LLS

Settore disciplinare:      INF/01 - Informatica

### Docente

Vincenzo Gervasi

*gervasi@di.unipi.it*  
*www.di.unipi.it/~gervasi*  
*stanza n.333, dip.to di Informatica*  
*tel:0502212773*

Ulteriore pagina web del corso: <http://circe.di.unipi.it/~gervasi/LLS05>

---

## Laboratorio di linguaggi di sistema B

Codice: AA536   Crediti: 3   Semestre: 1   Sigla: LLS

Settore disciplinare:            INF/01 - Informatica

### Docente

Vincenzo Gervasi  
*gervasi@di.unipi.it*  
*www.di.unipi.it/~gervasi*  
*stanza n.333, dip.to di Informatica*  
*tel:0502212773*

Ulteriore pagina web del corso: <http://circe.di.unipi.it/~gervasi/LLS05>

---

## Laboratorio di programmazione concorrente e di sistema A

Codice: AA538   Crediti: 6   Semestre: 2   Sigla: LCS

Settore disciplinare:            INF/01 - Informatica

### Docente

Susanna Pelagatti  
*susanna@di.unipi.it*  
*www.di.unipi.it/~susanna*  
*stanza n.346, dip.to di Informatica*  
*tel:0502212772*

### Prerequisiti

Si consiglia di aver seguito e superato l'esame di AA536 "Laboratorio di

## Linguaggi di Sistema''

### Obiettivi di apprendimento

Il corso intende fornire le conoscenze di base relative alla programmazione C con chiamate di sistema Unix/POSIX. In particolare, vengono prese in considerazione le chiamate standard per il trattamento di file e directory, processi, thread, segnali, pipe, socket e vari meccanismi di sincronizzazione.

*Conoscenze.* Lo studente acquisirà conoscenze sulla programmazione concorrente utilizzando processi e thread in ambiente Unix/Posix, con particolare riferimento ai sistemi Linux.

*Capacità.* Lo studente saprà progettare e realizzare un sistema client server di medie dimensioni utilizzando gli strumenti messi a disposizione dallo standard POSIX. Saprà inoltre selezionare all'interno dello standard le primitive più adeguate, in modo da ottenere un software robusto ed efficiente. Saprà infine codificare e sviluppare la soluzione progettata usando ANSI C e gli strumenti dell'ambiente GNU.

*Comportamenti.* Lo studente sarà cosciente delle problematiche principali legate all'utilizzo di C+POSIX su diverse piattaforme e saprà valutare indipendentemente le scelte progettuali più adeguate per ottenere del software portabile e corretto.

### Descrizione

Il corso e' diviso logicamente in due parti. La prima descrive sommariamente la shell di Unix, introduce gli script (con riferimento a Bash), descrive l'ambiente di sviluppo GNU ed i tool associati. La seconda tratta la programmazione concorrente e di sistema utilizzando le System Call POSIX. In particolare vengono descritte e sperimentate le SC relative ai file, ai processi, ai thread, ai segnali, alle pipe ed ai socket. Ogni argomento prevede lo svolgimento di esercizi in laboratorio da parte degli studenti.

### Programma

Il linguaggio usato nel corso e' ANSI C (lo studente deve già conoscerlo da LLS)

1. Introduzione. Gestione di progetti C complessi: make, cenni alla programmazione di shell (2h + 2h laboratorio)
2. Standard POSIX, system call, gestione degli errori (2h + 2h laboratorio)
3. Chiamate di sistema relative ai file: open, close, read, write, lseek(2h + 2h laboratorio)
4. SC avanzate sui file: stat, SC su directory (2h + 2h laboratorio)
5. Gestione di processi: fork, exec (2h + 2h laboratorio)
6. Gestione di thread: pthread\_create(), pthread\_join() etc (2h + 2h laboratorio)
7. IPC di base: semafori e mutex (2h + 2h laboratorio)
8. IPC di base: pipe con nome e pipe senza nome(2h + 2h laboratorio)
9. Gestione dei segnali (2h + 2h laboratorio)
10. Socket (2h + 2h laboratorio)

Ore lezione: 20 Ore esercitazione: 40

### **Bibliografia**

*Testo.* M.J. Rochkind Advanced Unix Programming. Addison Wesley 2004 Più materiale in linea accessibile dalla pagina web del corso.

*Software.* Postazioni Linux con ambiente GNU sui PC del laboratorio. Tutto il software utilizzato è open source e scaricabile gratuitamente dagli studenti da vari siti web oppure ottenibile in copia su CD direttamente dal centro di calcolo.

### **Modalità di esame**

La valutazione avverrà mediante lo sviluppo di un progetto didattico in gruppi di 1-2 studenti più una prova orale individuale. Il progetto verterà sullo sviluppo di un semplice sistema client server utilizzando gli strumenti presentati a lezione. Parti del progetto verranno svolte durante le ore di laboratorio con l'aiuto del docente.

Ulteriore pagina web del corso: <http://www.di.unipi.it/~susanna/LPC/> (sarà disponibile dall'inizio del corso: Febbraio 2006)

---

## **Laboratorio di programmazione concorrente e di sistema B**

Codice: AA538 Crediti: 6 Semestre: 2 Sigla: LCS

Settore disciplinare: INF/01 - Informatica

### **Docente**

Susanna Pelagatti  
*susanna@di.unipi.it*  
*www.di.unipi.it/~susanna*  
*stanza n. 346, dip. to di Informatica*  
*tel: 0502212772*

### **Prerequisiti**

Si consiglia di aver seguito e superato l'esame di AA536 "Laboratorio di Linguaggi di Sistema"

### **Obiettivi di apprendimento**

Il corso intende fornire le conoscenze di base relative alla programmazione C con chiamate di sistema Unix/POSIX. In particolare, vengono prese in considerazione le chiamate standard per il trattamento di file e directory, processi, thread, segnali, pipe, socket e vari meccanismi di sincronizzazione.

*Conoscenze.* Lo studente acquisirà conoscenze sulla programmazione

concorrente utilizzando processi e thread in ambiente Unix/Posix, con particolare riferimento ai sistemi Linux.

*Capacità.* Lo studente saprà progettare e realizzare un sistema client server di medie dimensioni utilizzando gli strumenti messi a disposizione dallo standard POSIX. Saprà inoltre selezionare all'interno dello standard le primitive più adeguate, in modo da ottenere un software robusto ed efficiente. Saprà infine codificare e sviluppare la soluzione progettata usando ANSI C e gli strumenti dell'ambiente GNU.

*Comportamenti.* Lo studente sarà cosciente delle problematiche principali legate all'utilizzo di C+POSIX su diverse piattaforme e saprà valutare indipendentemente le scelte progettuali più adeguate per ottenere del software portabile e corretto.

## **Descrizione**

Il corso e' diviso logicamente in due parti. La prima descrive sommariamente la shell di Unix, introduce gli script (con riferimento a Bash), descrive l'ambiente di sviluppo GNU ed i tool associati. La seconda tratta la programmazione concorrente e di sistema utilizzando le System Call POSIX. In particolare vengono descritte e sperimentate le SC relative ai file, ai processi, ai thread, ai segnali, alle pipe ed ai socket. Ogni argomento prevede lo svolgimento di esercizi in laboratorio da parte degli studenti.

## **Programma**

Il linguaggio usato nel corso e' ANSI C (lo studente deve già conoscerlo da LLS)

1. Introduzione. Gestione di progetti C complessi: make, cenni alla programmazione di shell (2h + 2h laboratorio)
2. Standard POSIX, system call, gestione degli errori (2h + 2h laboratorio)
3. Chiamate di sistema relative ai file: open, close, read, write, lseek(2h + 2h laboratorio)
4. SC avanzate sui file: stat, SC su directory (2h + 2h laboratorio)
5. Gestione di processi: fork, exec (2h + 2h laboratorio)
6. Gestione di thread: pthread\_create(), pthread\_join() etc (2h + 2h laboratorio)
7. IPC di base: semafori e mutex (2h + 2h laboratorio)
8. IPC di base: pipe con nome e pipe senza nome(2h + 2h laboratorio)
9. Gestione dei segnali (2h + 2h laboratorio)
10. Socket (2h + 2h laboratorio)

Ore lezione: 20 Ore esercitazione: 40

## **Bibliografia**

*Testo.* M.J. Rochkind Advanced Unix Programming. Addison Wesley 2004 Più materiale in linea accessibile dalla pagina web del corso.

*Software.* Postazioni Linux con ambiente GNU sui PC del laboratorio. Tutto il software utilizzato è open source e scaricabile gratuitamente dagli studenti da vari siti web oppure ottenibile in copia su CD direttamente dal centro di calcolo.

## Modalità di esame

La valutazione avverrà mediante lo sviluppo di un progetto didattico in gruppi di 1-2 studenti più una prova orale individuale. Il progetto verterà sullo sviluppo di un semplice sistema client server utilizzando gli strumenti presentati a lezione. Parti del progetto verranno svolte durante le ore di laboratorio con l'aiuto del docente.

Ulteriore pagina web del corso: <http://www.di.unipi.it/~susanna/LPC/> (sarà disponibile dall'inizio del corso: Febbraio 2006)

---

## Laboratorio di programmazione di rete A

Codice: AA018   Crediti: 6   Semestre: 1   Sigla: LPR

Settore disciplinare:            INF/01 - Informatica

### Docente

Laura Ricci  
*ricci@di.unipi.it*  
*www.di.unipi.it/~ricci*  
*stanza n.348, dip.to di Informatica*  
*tel: 0502212768*

### Prerequisiti

Laboratori del primo e secondo anno, conoscenza del linguaggio di programmazione Java

### Obiettivi di apprendimento

Fornire le basi sulle tecniche di programmazione di applicazioni di rete con gli strumenti tipici dell'ambiente Internet/Java (Socket, RMI). Il corso si propone di fornire le conoscenze di base necessarie per la realizzazione di (semplici) applicazioni di rete (applicazioni WEB, per file transfer, di posta elettronica, chat, etc.) In particolare, si forniranno le conoscenze necessarie per l'utilizzo di Socket (indirizzi, socket TCP, socket UDP, modelli connection-oriented e connectionless, comunicazioni punto a punto e multicast/broadcast, modello client/server). Verranno inoltre presentate le caratteristiche base di modelli di programmazione piu' ad alto livello, tipicamente RMI/CORBA/RPC. In particolare verranno svolti esercizi utilizzando lo RMI primitivo di Java.

### Programma

Parte prima: socket

- Modello client/server, server seriali e concorrenti.
- Socket (lato cliente e lato servente) con TCP con indirizzi visti come coppie

- nomehost,numeroporta.
  - o Accesso a servizi standard
  - o Realizzazione di semplici server concorrenti e non
- Indirizzi Inet
- UDP
  - o Accesso a servizi standard
  - o Port scanning
  - o Utilizzo di multicast
  - o Implementazione di livelli diversi di affidabilita' sopra UDP

Parte seconda: RMI

- RMI
- relazioni con CORBA/RPC
- Utilizzo di RMI per la realizzazione di semplici servizi di rete

Parte terza: Progetto

- Assegnazione e discussione in aula del progetto conclusivo del corso

Ore lezione: 24 Ore esercitazione: 24

#### Bibliografia

- Harold *JAVA Network Programming 2nd edition* O'Reilly 2001 (ne esiste anche una versione tradotta in italiano presso Jackson, relativa alla prima edizione del testo. Sebbene meno completa, copre praticamente tutti gli argomenti trattati nel corso).
- *consultazione* Huges et al. *JAVA Network Programming* Manning (il testo e' da considerarsi come un manuale da tenere sulla scrivania durante la realizzazione di applicazioni di rete)

#### Modalità di esame

Scritto e orale

Ulteriore pagina web del corso: <http://www.di.unipi.it/%7Ericci/labprogrete0506.html>

---

## Laboratorio di programmazione di rete B

Codice: AA018 Crediti: 6 Semestre: 1 Sigla: LPR

Settore disciplinare: INF/01 - Informatica

#### Docente

Gianluigi Ferrari  
[giangi@di.unipi.it](mailto:giangi@di.unipi.it)

*www.di.unipi.it/~giangi*  
*stanza n. 358, dip.to di Informatica*  
*tel: 0502212766*

## **Prerequisiti**

Laboratori del primo e secondo anno, conoscenza del linguaggio di programmazione Java.

## **Obiettivi di apprendimento**

Il laboratorio si pone l'obiettivo di fornire le conoscenze di base necessarie per la progettazione e realizzazione di (semplici) applicazioni di rete (applicazioni client-server applicazioni peer-to-peer). Il laboratorio pone una enfasi particolare nella sperimentazione di paradigmi per la programmazione di applicazioni di rete

*Conoscenze.* Lo studente al termine del laboratorio acquisirà le conoscenze delle tecniche di base per programmare applicazioni di rete sia tradizionali (client-server) che innovative (peer-to-peer) in termini di comunicazione via socket (socket TCP, socket UDP e Multicast) e programmazione con oggetti remoti.

*Capacità.* Lo studente al termine del laboratorio è in grado di utilizzare le gli strumenti e le tecnologie di base (socket e oggetti remoti) di supporto al progetto, realizzazione e testing di applicazioni di rete non banali.

*Comportamenti.* Lo studente al termine del laboratorio ha acquisito un livello di autonomia che permette di valutare le tipologie differenti di applicazioni di rete e le tecnologie necessarie per il loro progetto e realizzazione. Sarà inoltre cosciente delle problematiche inerenti al testing di applicazioni di rete in contesti operativi.

## **Indicazioni metodologiche**

Per conseguire gli obiettivi indicati, è necessario:

- organizzare il processo di apprendimento in moduli flessibili, posti in sequenza logica;
- presentare in modo approfondito i diversi paradigmi di programmazione
- presentare in modo approfondito i diversi meccanismi di programmazione
- evidenziare i limiti delle tecniche presentate in modo da favorire il formarsi di una capacità critica nello studente;
- coordinarsi con i docenti dei corsi indicati nei prerequisiti e dei corsi paralleli del semestre;
- proporre l'analisi di casi di studio tratti da esperienze reali riportate in letteratura o derivanti da ricerche del docente;
- separare gli aspetti e le scelte puramente tecnologiche da quelle funzionali alla soluzione di un problema di natura applicativa.

## **Programma**

Il corso si pone l'obiettivo di fornire le conoscenze necessarie per l'utilizzo di

Socket (indirizzi, URL, socket TCP e UDP, modelli connection- oriented e connectionless, comunicazioni punto a punto, multicast e broadcast, modello client/server). Saranno inoltre presentate le caratteristiche base di modelli di programmazione più ad alto livello: RMI, CORBA ed RPC. In particolare, verranno svolti esercizi utilizzando Java RMI

- Introduzione: La programmazione di applicazioni di rete (2 ore)
- Paradigmi per la programmazione di applicazioni di rete (4 ore)
- Programmazione con socket TCP e con socket UDP: (16 ore)
- Middleware di programmazione ad oggetti: Remote Method Invocation (RMI) (16 ore)
- Applets e Security (4 Ore)
- Assegnazione e discussione del progetto conclusivo del corso (4 ore)

Ore lezione: 24 Ore esercitazione: 24

### Bibliografia

Harold, JAVA Network Programming Third edition (O'Reilly). 2005  
Hughes et al. JAVA Network Programming 2nd edition (Manning). 2001  
Dispense e note didattiche distribuite dal docente.

### Modalità di esame

Realizzazione e discussione di un progetto di programmazione. Il progetto di programmazione è un caso di studio derivato da casi di studio e/o applicazioni reali.

Ulteriore pagina web del corso:

<http://www.di.unipi.it/~marcod/Didattica/LabReti/index.html>

---

## Laboratorio di programmazione di strutture dati A

Codice: AA537 Crediti: 3 Semestre: 1 Sigla: LSD

Settore disciplinare: INF/01 - Informatica

### Docente

Chiara Bodei  
*chiara@di.unipi.it*  
*www.di.unipi.it/~chiara*  
*stanza n. 322, dip.to di Informatica*  
*tel: 0502212760*

### Prerequisiti

Il corso è la naturale prosecuzione del corso di **Laboratorio di Introduzione**

**alla Programmazione.** Avere sostenuto questo esame non è un prerequisito formale. Prequisito SOSTANZIALE è essere in grado di editare, compilare ed eseguire un programma Java su una piattaforma a scelta con uno strumento di sviluppo a scelta (preferibilmente Eclipse).

### Obiettivi di apprendimento

Il corso ha lo scopo di introdurre gli studenti ai principi di base della programmazione ad oggetti, mediante l'uso di JAVA. Le conoscenze acquisite vengono poi utilizzate per la risoluzione di semplici problemi connessi all'uso di alcune **strutture dati elementari** (in particolare **pile**, **code**, **liste** e **alberi**).

*Conoscenze.* la teoria delle strutture dati

*Capacità.* programmazione Java di strutture dati

*Comportamenti.* capacità di impostare e mettere a punto indipendentemente programmi Java di media complessità

### Descrizione

Il corso si articola in un'ora di lezione teorica e due ore di laboratorio alla settimana. Verrà data particolare enfasi alla pratica della programmazione ad oggetti e alla documentazione del codice prodotto.

La frequenza è obbligatoria per chi non ha mai frequentato il vecchio corso di LSD da 4 CFU. Sono esenti dall'obbligo gli studenti lavoratori che sono tuttavia pregati contattare con il docente all'inizio delle lezioni.

### Indicazioni metodologiche

Il corso si articola in un'ora di lezione teorica e due ore di laboratorio alla settimana. Verrà data particolare enfasi alla pratica della programmazione ad oggetti e alla documentazione del codice prodotto.

### Programma

- Tipi di dato astratto e interfacce
- Strutture dati lineari
  - Pile
  - Code
  - Liste
- Alberi binari e Alberi binari di ricerca
- Insiemi

Ore lezione: 12 Ore esercitazione: 24

### Bibliografia

- Materiale distribuito dai docenti
- Cay S. Horstmann, [\*Concetti di informatica e fondamentali di JAVA 2\*](#), Apogeo.
- Bruce Eckel, *Thinking in Java II ed (in italiano)*, Apogeo, 2004.
- Bruce Eckel, [\*Thinking in Java \(in inglese, gratuito, online\)\*](#)

## Modalità di esame

Agli studenti verrà assegnato un **progetto**. Il progetto dovrà essere completato e consegnato al docente **entro il 20 settembre 2006**. Il voto del corso verrà dato in base alla valutazione del progetto.

Ulteriore pagina web del corso:

[http://www.cli.di.unipi.it/~bodei/CORSO\\_06/LSD/index.html](http://www.cli.di.unipi.it/~bodei/CORSO_06/LSD/index.html)

---

## Laboratorio di programmazione di strutture dati B

Codice: AA537   Crediti: 3   Semestre: 1   Sigla: LSD

Settore disciplinare:            INF/01 - Informatica

### Docente

Francesco Romani  
*romani@di.unipi.it*  
*www.di.unipi.it/~romani*  
*stanza n.326, dip.to di Informatica*  
*tel:0502212734*

### Prerequisiti

Il corso è la naturale prosecuzione del corso di **Laboratorio di Introduzione alla Programmazione**. Avere sostenuto questo esame non è un prerequisito formale. Prequisito SOSTANZIALE è essere in grado di editare, compilare ed eseguire un programma Java su una piattaforma a scelta con uno strumento di sviluppo a scelta (preferibilmente Eclipse).

### Obiettivi di apprendimento

Il corso ha lo scopo di introdurre gli studenti ai principi di base della programmazione ad oggetti, mediante l'uso di JAVA. Le conoscenze acquisite vengono poi utilizzate per la risoluzione di semplici problemi connessi all'uso di alcune **strutture dati elementari** (in particolare **pile**, **code**, **liste** e **alberi**).

*Conoscenze.* la teoria delle strutture dati

*Capacità.* programmazione Java di strutture dati

*Comportamenti.* capacità di impostare e mettere a punto indipendentemente programma Java di media complessità

### Descrizione

Il corso si articola in un'ora di lezione teorica e due ore di laboratorio alla settimana. Verrà data particolare enfasi alla pratica della programmazione ad oggetti e alla documentazione del codice prodotto.

La frequenza è obbligatoria per chi non ha mai frequentato il vecchio corso di LSD da 4 CFU. Sono esenti dall'obbligo gli studenti lavoratori che sono tuttavia pregati contattare con il docente all'inizio delle lezioni.

### Indicazioni metodologiche

Il corso si articola in un'ora di lezione teorica e due ore di laboratorio alla settimana. Verrà data particolare enfasi alla pratica della programmazione ad oggetti e alla documentazione del codice prodotto.

### Programma

- Tipi di dato astratto e interfacce
- Strutture dati lineari
  - Pile
  - Code
  - Liste
- Alberi binari e Alberi binari di ricerca
- Insiemi

Ore lezione: 12 Ore esercitazione: 24

### Bibliografia

- Materiale distribuito dai docenti
- Cay S. Horstmann, *Concetti di informatica e fondamentali di JAVA 2*, Apogeo.
- Bruce Eckel, *Thinking in Java II ed (in italiano)*, Apogeo, 2004.
- Bruce Eckel, *Thinking in Java (in inglese, gratuito, online)*

### Modalità di esame

Agli studenti verrà assegnato un **progetto**. Il progetto dovrà essere completato e consegnato al docente **entro il 20 settembre 2006**. Il voto del corso verrà dato in base alla valutazione del progetto.

Ulteriore pagina web del corso:

[http://www.cli.di.unipi.it/~romani/CORSO\\_06/LSD/index.html](http://www.cli.di.unipi.it/~romani/CORSO_06/LSD/index.html)

---

## Lingua inglese

Codice: nw200 Crediti: 0 Semestre: 2 Sigla: ING

Settore disciplinare: L-LIN/12 - Lingua e traduzione - Lingua Inglese

Docente

Emily Fullwood

*fullwood@di.unipi.it*  
*www.di.unipi.it/~fullwood*  
*tel: 0502212700*

## **Descrizione**

Il corso e' di livello pre-intermedio ed ha una durata di 40 ore.

Al corso possono partecipare un massimo di 100 studenti iscritti al corso di laurea triennale in Informatica (che devono quindi essere iscritti in una apposita lista).

Il corso non ha un esame finale e non da diritto a maturare crediti formativi universitari.

Per ottenere i 3 cfu di "Conoscenza della lingua inglese - LL001", gli studenti devono comunque superare il test offerto dal CLI.

---

## **Linguaggi**

Codice: AA029   Crediti: 6   Semestre: 1   Sigla: LIN

Settore disciplinare:            INF/01 - Informatica

### **Docente**

Giorgio Levi  
*levi@di.unipi.it*  
*www.di.unipi.it/~levi*  
*stanza n.281, dip.to di Informatica*  
*tel: 0502212746*

### **Prerequisiti**

Nessuno

### **Obiettivi di apprendimento**

Analizzare i principi ed i meccanismi che stanno alla base del progetto, della realizzazione e dell'uso dei linguaggi di programmazione. Esaminare i principali paradigmi di programmazione e le loro tecniche di realizzazione. Sviluppare una capacità critica che permetta di valutare l'adeguatezza di un paradigma (e di un linguaggio) per la soluzione di uno specifico problema applicativo.

### **Descrizione**

Si introdurranno i costrutti principali dei linguaggi imperativi, funzionali e orientati ad oggetti e si svilupperanno varie implementazioni (interpretative e compilative) per raffinamenti successivi (usando metodi formali quali la

trasformazione di ricorsione in iterazione, l'eliminazione dell'ordine superiore, l'interpretazione astratta e la valutazione parziale) a partire da una semantica formale espressa nel linguaggio ML.

## Programma

1. Macchine astratte, linguaggi, interpretazione e compilazione (2 ore).
2. Richiami di semantica, ML come metalinguaggio (3 ore).
3. Tipi di dato, tipi e controllo dei tipi (4 ore).
4. Controllo di sequenza, procedure, ricorsione (5 ore).
5. Ambiente, scoping, parametri, moduli, classi (6 ore).
6. Analisi statica (3 ore).
7. Gestione della memoria (1 ora).
8. Specializzazione, traduzione e strutture a run time (2 ore).

## Bibliografia

T.W. Pratt & M.V. Zelkowitz, Programming languages. Design and implementation. Prentice-Hall 1996.

## Modalità di esame

Scritto e orale

Ulteriore pagina web del corso:

<http://www.di.unipi.it/~levi/corsoLP/pagina.html>

---

## Linguaggi e calcolabilità

Codice: AA030   Crediti: 6   Semestre: 1   Sigla: LC

Settore disciplinare:            INF/01 - Informatica

### Docente

Egon Börger  
*boerger@di.unipi.it*  
*www.di.unipi.it/~boerger*  
*stanza n.289, dip.to di Informatica*  
*tel: 0502212730*

### Prerequisiti

Laurea triennale di informatica

### Obiettivi di apprendimento

L'obiettivo del corso e' di fornire allo studente una padronanza di metodi di specifica e di analisi di sistemi software complessi. Si studiano i principali modelli di computazione insieme ai metodi fondamentali per la definizione, la verifica e la validazione di architetture software.

## **Descrizione**

Modelli di computazione: Funzioni Ricorsive e le loro limitazioni (insolubilita e diagonazzizzazione), computazioni su tipi di dati astratti (Abstract State Machines). Elementi di semantica di linguaggi (sistemi di tipi, analisi statica). Metodi di specifica astratta: modelli ASM per linguaggi ad alto livello (Java,C#), per macchine virtuali (Java Virtual Machine, CLR), per architetture software ed servizi web.

## **Programma**

vedi descrizione del corso

## **Bibliografia**

Articoli segnalati durante il corso ed il seguente testo di base:

Abstract State Machines (by E. Boerger and R. Staerk, Springer Pu.Co. 2003)

Per Java verra usato anche il testo:

Java and the Java Virtual Machine — Definition, Verification, Validation

by R.Staerk, J. Schmid, Egon Boerger (Springer Pu.Co. 2001)

## **Modalità di esame**

Scritto e orale.

Alternativa: 2 compiti scritti alla fine di ottobre/dicembre con possibilita di recupero al primo appello di gennaio di uno dei due compiti andato male.

Ulteriore pagina web del corso: <http://www.di.unipi.it/~boerger/teaching.html>

---

## **Linguaggio e metodi della matematica A**

Codice: AA004   Crediti: 6   Semestre: 1   Sigla: LMM

Settore disciplinare: MAT/01 - Logica Matematica

#### Docente

Massimo Caboara  
*caboara@dm.unipi.it*  
*www.dima.unige.it/~caboara/*  
Tel. 050/2213283

#### Obiettivi di apprendimento

L'obiettivo del corso è duplice. Innanzitutto si vuole far acquisire allo studente una dimestichezza con le principali notazioni logico-insiemistiche (connettivi, insiemi, funzioni), e con i principali metodi dimostrativi (tra cui l'induzione matematica). In secondo luogo si vogliono comunicare precisi contenuti matematici nell'ambito della matematica del discreto (aritmetica, calcolo combinatorio). L'integrazione dei due aspetti, linguistici e contenutistici, è ritenuta indispensabile per il conseguimento di questi obiettivi.

#### Descrizione

Insiemi, relazioni, funzioni. Logica e tecniche di dimostrazione. Induzione, aritmetica, combinatorica.

#### Programma

Nozioni di calcolo proposizionale e calcolo dei predicati. Insiemistica elementare, funzioni, relazioni. Metodi dimostrativi. Induzione. Aritmetica e aritmetica modulare. Combinatoria elementare.

#### Bibliografia

Peter J. Eccles, "An introduction to mathematical reasoning", Cambridge University Press. Kenneth H. Rosen, "Discrete mathematics and its applications", McGraw Hill.

Ulteriore pagina web del corso: <http://www.dm.unipi.it/~caboara>

---

## Linguaggio e metodi della matematica B

Codice: AA004   Crediti: 6   Semestre: 1   Sigla: LMM

Settore disciplinare: MAT/01 - Logica Matematica

#### Docente

Eugene Stepanov  
*e.stepanov@sns.it*

### **Prerequisiti**

nessuno

### **Obiettivi di apprendimento**

Introdurre il linguaggio e le tecniche di dimostrazione della matematica, sviluppando come esempio alcuni temi fondamentali dell'aritmetica e della combinatorica.

### **Descrizione**

Elementi di logica elementare: connettivi logici, proposizioni ed enunciati. Insiemi, relazioni, grafi, funzioni. Logica e tecniche di dimostrazione. Induzione, aritmetica, combinatorica.

### **Programma**

Proposizioni ed enunciati; connettivi booleani, tavole di verità, quantificatori. Insiemi, prodotto cartesiano, insieme potenza. Funzioni iniettive, surgettive, bigettive. Inversa di una funzione bigettiva. Numeri naturali e principio di induzione. Tecniche di dimostrazione per induzione. Definizioni per ricorrenza. ricursione. Elementi di calcolo combinatorio: numero di elementi del prodotto cartesiano di insiemi finiti, coefficiente binomiale, principio di inclusione-esclusione. Divisione euclidea ed algoritmo di Euclide per il massimo comune divisore fra due interi. Numeri primi e teorema di fattorizzazione unica. L'equazione diofantea  $ax+by=c$ : risolubilità e determinazione delle soluzioni. Definizione di congruenza modulo  $m$ . Risoluzione delle equazioni lineari con congruenze. Il teorema cinese del resto. Il piccolo teorema di Fermat.

### **Bibliografia**

Kenneth H. Rosen, Discrete mathematics and its applications, Mc Graw-Hill.  
Peter J. Eccles, An Introduction to Mathematical Reasoning, Cambridge University Press.  
Lindsay Childs, Algebra: un'introduzione concreta, ETS.

### **Modalità di esame**

Scritto e orale

---

## **Linguaggio e metodi della matematica C**

Codice: AA004   Crediti: 6   Semestre: 1   Sigla: LMM

Settore disciplinare:            MAT/01 - Logica Matematica

## Docente

Giovanni Gaiffi  
*gaiffi@dm.unipi.it*  
*www.dm.unipi.it/~gaiffi/*  
*Tel. 050-2213242*

## Obiettivi di apprendimento

Introdurre il linguaggio e le tecniche di dimostrazione della matematica, sviluppando come esempio alcuni temi fondamentali dell'aritmetica e del calcolo combinatorio.

## Descrizione

\* Primi concetti di logica e discussione di alcune tecniche di dimostrazione. \* Insiemi, relazioni, funzioni, cardinalità del discreto e del continuo \* Aritmetica, calcolo combinatorio.

## Programma

Proposizioni e connettivi, tavole di verità, formalizzazione, equivalenza logica, quantificatori. Insiemi, funzioni, prodotto cartesiano, insieme potenza. Numeri naturali, induzione, calcolo combinatorio e coefficienti binomiali. Divisibilità, numeri primi, massimo comun divisore e teorema di Bezout, rappresentazione di un numero in una data base. Congruenze, piccolo teorema di Fermat. Cardinalità, insiemi numerabili, cardinalità dei numeri reali, principio dei cassetti, formula di inclusione-esclusione. Operazioni, relazioni, relazioni d'ordine, relazioni di equivalenza, grafi, alberi.

## Bibliografia

Per prima cosa si consiglia di fare riferimento agli appunti delle lezioni e alle dispense integrative che verranno messe a disposizione nella pagina web del corso. Le lezioni saranno comunque molto vicine al testo: PETER J. ECCLES - AN INTRODUCTION TO MATHEMATICAL REASONING, Cambridge. Come vedete si tratta di un testo in inglese: vorremmo che questo fosse di stimolo per prendere confidenza con l'inglese scientifico (si tratta di una acquisizione necessaria e importante per i vostri studi). Per non lasciare senza nessun riferimento coloro che dell'inglese proprio non ne vogliono sapere (male, male...), segnalo un testo in italiano: BIANCHI, GILLIO - INTRODUZIONE ALLA MATEMATICA DISCRETA 2a Ed., McGraw-Hill che contiene anche argomenti che verranno trattati nel corso di algebra.

## Modalità di esame

Scritto e orale

Ulteriore pagina web del corso: [http://www.dm.unipi.it/~gaiffi/lmm\\_2005/](http://www.dm.unipi.it/~gaiffi/lmm_2005/)

---

## Logistica

Codice: AA043   Crediti: 6   Semestre: 1   Sigla: LOG

Settore disciplinare:            MAT/09 - Ricerca Operativa

### Docente

Maria Grazia Scutellà  
*scut@di.unipi.it*  
*www.di.unipi.it/~scut*  
*stanza n.363, dip.to di Informatica*  
*tel:0502212771*

### Prerequisiti

Si consiglia la frequenza del corso di Ricerca Operativa

### Obiettivi di apprendimento

Il corso si propone di presentare la struttura ed il funzionamento dei Sistemi logistici. Tali caratteristiche verranno esemplificati mediante fogli elettronici (spreadsheets).

*Conoscenze.* Obiettivo del corso è l'insegnamento della struttura e del funzionamento di una catena logistica, con enfasi sul settore produttivo e dei servizi. Al termine del corso lo studente conoscerà alcune caratteristiche basilari relative alla struttura ed al funzionamento di un sistema logistico. Specificatamente, conoscerà le principali problematiche, e le relative tecniche di modellazione e di risoluzione, nell'ambito degli aspetti di progetto (localizzazione di facilities,, routing di prodotti e di informazioni, e problematiche congiunte di localizzazione-routing) e di gestione (funzionamento dei centri di distribuzione, gestione delle scorte, previsione della domanda) di un sistema logistico. Tali problematiche verranno formalizzate ed analizzate facendo ricorso a strumenti di Programmazione Matematica. Si tratta di conoscenze altamente qualificanti, in grado di agevolare l'inserimento dello studente nel mondo del lavoro.

*Capacità.* Al termine del corso lo studente sarà in grado di formalizzare alcuni problemi decisionali relativi al progetto e gestione di un sistema logistico mediante modelli di Programmazione Lineare Intera. Sarà in grado di risolvere tali modelli mediante un solutore (Excel), analizzando i risultati ottenuti. Sarà inoltre in grado di risolvere alcuni problemi decisionali tipici della logistica mediante algoritmi "ad-hoc". Lo studente acquisirà quindi competenze a livello strategico-decisionale, che gli permetteranno di affrontare, valutare e risolvere importanti problemi decisionali a livello di sistemi logistici nel medio e nel lungo periodo.

*Comportamenti.* Al termine del corso lo studente sarà in grado di analizzare e valutare problemi decisionali in modo critico, autonomo e rigoroso, proponendo modelli di Programmazione Lineare Intera e/o procedure risolutive, per una

rigorosa comprensione e trattazione del problema decisionale posto.

### Descrizione

- Struttura e funzionamento dei Sistemi Logistici: la catena logistica, obiettivi di gestione;
- Metodi di previsione della domanda: metodi causali, metodi basati sulle serie temporali;
- Progetto della struttura logistica di un'impresa e problemi di localizzazione dei nodi logistici.
- Gestione delle scorte: gestione di un punto di stoccaggio; gestione di più punti di stoccaggio.
- Progetto e gestione dei Centri di distribuzione: dimensionamento delle aree di stoccaggio; dimensionamento dei lotti; problemi di caricamento dei mezzi di trasporto.
- Esempificazioni attraverso 'case studies': fogli elettronici.

### Programma

- Introduzione (2 ore): Struttura e funzionamento dei sistemi logistici: la catena logistica, obiettivi di gestione
- Problemi di localizzazione (10 ore): Modelli di localizzazione ad un livello con domanda deterministica e costante, modello di localizzazione a due livelli nel caso di più prodotti, modelli di copertura; modelli di p-centro
- Problemi di trasporto (routing) (6 ore): Modelli per il trasporto a lunga distanza (problemi di progetto di rete); modelli per il trasporto a breve distanza
- Problemi di localizzazione e routing (4 ore): Modello base per localizzazione e routing
- Problemi di gestione delle scorte (8 ore): Gestione di un punto di stoccaggio (singolo prodotto e domanda deterministica e costante), gestione di un punto di stoccaggio con più prodotti; gestione di un punto di stoccaggio con domanda aleatoria; gestione di più punti di stoccaggio; gestione di articoli a bassa domanda
- Problemi di progetto e gestione di centri di distribuzione (8 ore): Progetto di un centro di distribuzione; dimensionamento delle aree di stoccaggio; dimensionamento dei lotti
- Metodi di previsione della domanda (6 ore): Metodi causali; metodi basati sulle serie temporali (andamento tendenziale costante, lineare ed effetto stagionale)
- Esempificazioni attraverso "case studies" (4 ore): fogli elettronici

Ore lezione: 32    Ore esercitazione: 12    Ore laboratorio: 4

### Bibliografia

- G. Ghiani, R. Musmanno, "Modelli e Metodi per l'Organizzazione dei Sistemi Logistici, Pitagora Ed. 2000 - Appunti di Logistica (reperibili dal sito web del corso)
- Esempi di modellazione mediante Excel (reperibili dal sito web del corso) - Software: solutore Excel

## Modalità di esame

Progetto (formulazione e risoluzione di un problema logistico mediante il software Excel) seguito da una prova orale

Ulteriore pagina web del corso:

<http://www.di.unipi.it/di/groups/optimize/ORGroup.html>

---

## Matematica computazionale

Codice: AA039 Crediti: 6 Semestre: 2 Sigla: MC

Settore disciplinare: MAT/08 - Analisi Numerica

### Docente

Luca Gemignani  
*gemignan@dm.unipi.it*  
*www.dm.unipi.it/~gemignan/*  
*Tel. 050-2213291*

### Obiettivi di apprendimento

Apprendimento delle tecniche e degli strumenti per la risoluzione numerica di problemi discreti tipicamente di algebra lineare che scaturiscono nelle applicazioni della matematica.

### Descrizione

Il corso descrive i principali metodi numerici per la risoluzione approssimata di problemi di algebra lineare che intervengono nelle applicazioni della matematica. Enfasi è posta sulle problematiche che scaturiscono dall'implementazione degli algoritmi e dalla validazione dei risultati.

### Programma

1. Metodi diretti per sistemi lineari
  - o Condizionamento e risoluzione di sistemi triangolari (Par. 4.1, 4.2)
  - o Matrici elementari e fattorizzazione LU (Par. 4.4 e 4.5)
  - o Matrici elementari di Householder e fattorizzazione QR (Par. 4.12 e 4.13)
2. Metodi iterativi per sistemi lineari
  - o Generalità e convergenza dei metodi iterativi (Par. 5.1, 5.2 e 5.3)
  - o Metodo del gradiente coniugato (Par. 5.7 { teoremi solo enunciati)
3. Metodi numerici per il calcolo di autovalori ed autovettori
  - o Richiami di algebra lineare. La forma canonica di Schur.
  - o Condizionamento (Bauer-Fike, Par. 6.2)
  - o Introduzione generale ai metodi (Par. 6.4)
  - o Riduzione di una matrice simmetrica in forma tridiagonale con il

- metodo di Householder (Par. 6.5)
  - o Calcolo degli autovalori con la successione di Sturm (Par. 6.6)
  - o Metodo delle potenze e potenze inverse (Par. 6.10 e 6.11)
  - o Metodi Divide-et-Impera per matrici tridiagonali simmetriche (dispense distribuite)
4. Il problema lineare ai minimi quadrati
- o Formulazione del problema e metodo delle equazioni normali (Par. 7.1)
  - o Metodo QR per il calcolo della soluzione del problema ai minimi quadrati (Par. 7.2)
  - o Decomposizione ai valori singolari (SVD): esistenza della fattorizzazione e proprieta' (solo enunciati, Par. 7.4)
  - o Inversa generalizzata e risoluzione del problema ai minimi quadrati mediante la SVD (Par. 7.5 e 7.6)

### Bibliografia

- D. A. Bini, M. Capovani, O. Menchi, "Metodi Numerici per l'Algebra Lineare", Zanichelli Editore
- J. W. Demmel, "Applied Numerical Linear Algebra", SIAM, Philadelphia, 1997.

### Modalità di esame

Prova orale

Ulteriore pagina web del corso: <http://virmap.unipi.it/cgi-bin/virmap/regibo?registri:16723275;main>

---

## Matematica Computazionale: Laboratorio

Codice: AA040   Crediti: 6   Semestre: 2   Sigla: MCL

Settore disciplinare:      MAT/08 - Analisi Numerica

### Docente

Enrico Bozzo  
*bozzo@di.unipi.it*  
*www.di.unipi.it/~bozzo*  
*stanza n. 327, dip.to di Informatica*  
*tel: 0502212739*

### Obiettivi di apprendimento

Introdurre lo studente alla realizzazione di software numerico

## Descrizione

Il corso si propone di descrivere, implementare e sperimentare alcuni importanti algoritmi numerici.

## Programma

- Introduzione a Matlab
- Algoritmi per il problema lineare dei minimi quadrati.
- Il metodo del gradiente coniugato.
- La FFT ed alcune applicazioni.

## Bibliografia

- C. Moler, Numerical computing with MATLAB, <http://www.mathworks.com/moler/>, 2004.
- D. Higham, N. Higham, MATLAB Guide, SIAM, Philadelphia, PA, 2000.

## Modalità di esame

Scritto e orale

Ulteriore pagina web del corso: <http://www.di.unipi.it/~bozzo/mclnew.html>

---

## Metodologie di programmazione A

Codice: AA013   Crediti: 6   Semestre: 2   Sigla: MP

Settore disciplinare:      INF/01 - Informatica

### Docente

Marco Bellia  
*bellia@di.unipi.it*  
*www.di.unipi.it/~bellia*  
*stanza n.371, dip.to di Informatica*  
*tel:0502212781*

### Prerequisiti

corso di Fondamenti di Programmazione

### Obiettivi di apprendimento

Descrizione del modello di calcolo e delle tecniche di programmazione object-oriented con subtyping e polimorfismo.

*Conoscenze.* Conoscenza dei principi della programmazione a oggetti.

*Capacità.* Capacità di costruire programmi a oggetti in modo sistematico, di documentarli e di provarne le proprietà.

*Comportamenti.* Saper costruire applicazioni anche complesse sviluppando nuovi moduli, estendendo moduli esistenti ed integrandoli.

### **Indicazioni metodologiche**

Il corso prevede lezioni ed esercitazioni che mettano in pratica i principi esposti.

### **Programma**

Interpretazione, compilazione, supporto a run-time.

Oggetti e astrazione procedurale in Java (richiami).

Semantica operativa di un nucleo orientato ad oggetti con metodi, classi ed ereditarietà.

Metodologie per lo sviluppo di programmi sequenziali in Java

Eccezioni.

Astrazioni sui dati

Astrazioni sul controllo (iterazione)

Gerarchie di tipi

Astrazioni polimorfe.

Testing e debugging.

Ore lezione: 32 Ore esercitazione: 16

### **Bibliografia**

*Libro di testo:* B.Liskov and J. Guttag, Program Development in Java. Abstraction, Specification and Object-Oriented Design, Addison-Wesley, 2001.  
*Materiale* preparato durante il corso e disponibile sulla pagina web

### **Modalità di esame**

Compitini ed esami consistenti in uno scritto e un orale.

Ulteriore pagina web del corso:

<http://www.di.unipi.it/~bellia/METODOLOGIE/Anno05/Methodologie2005.htm>

Codice: AA013 Crediti: 6 Semestre: 2 Sigla: MP

Settore disciplinare: INF/01 - Informatica

### Docente

Giorgio Levi  
*levi@di.unipi.it*  
*www.di.unipi.it/~levi*  
*stanza n.281, dip.to di Informatica*  
*tel: 0502212746*

### Prerequisiti

corso di Fondamenti di Programmazione

### Obiettivi di apprendimento

Descrizione del modello di calcolo, delle tecniche di programmazione object-oriented con subtyping e polimorfismo, delle relative tecniche di specifica e di ragionamento.

*Conoscenze.* Conoscenza dei principali meccanismi di astrazione, dei principi della programmazione a oggetti e delle tecniche per ragionare sui programmi.

*Capacità.* Capacità di costruire programmi a oggetti in modo sistematico, di specificarli, documentarli e di provarne le proprietà.

*Comportamenti.* Saper costruire applicazioni anche complesse sviluppando nuovi moduli, estendendo moduli esistenti ed integrandoli.

### Indicazioni metodologiche

Il corso prevede lezioni ed esercitazioni che mettono in pratica i principi esposti.

### Programma

- Interpretazione, compilazione, supporto a run-time
- Oggetti e astrazione procedurale in Java (richiami)
- Semantica operativa di un nucleo orientato ad oggetti con metodi, classi ed ereditarietà
- Metodologie per lo sviluppo di programmi sequenziali in Java
  - Eccezioni
  - Astrazioni sui dati
  - Astrazioni di controllo (iterazione)
  - Gerarchie di tipi
  - Astrazioni polimorfe
- Testing e debugging

Ore lezione: 32 Ore esercitazione: 16

### Bibliografia

*Libro di testo:* B.Liskov and J. Guttag, Program Development in Java. Abstraction, Specification and Object-Oriented Design, Addison-Wesley, 2001.

Lucidi del corso reperibili sulla pagina web. Testi e correzioni dei compiti e degli esercizi di esame reperibili sulla pagina web.

### **Modalità di esame**

Prove di verifica intermedia, esami consistenti uno scritto e un orale.

Ulteriore pagina web del corso:

<http://www.di.unipi.it/~levi/corsoMP/pagina.html>

---

## **Metodologie di programmazione C**

Codice: AA013   Crediti: 6   Semestre: 2   Sigla: MP

Settore disciplinare:            INF/01 - Informatica

### **Docente**

Andrea Maggiolo Schettini  
*maggiolo@di.unipi.it*  
*www.di.unipi.it/~maggiolo*  
*stanza n.279, dip.to di Informatica*  
*tel:0502212759*

### **Prerequisiti**

corso di Fondamenti di Programmazione

### **Obiettivi di apprendimento**

Descrizione del modello di calcolo e delle tecniche di programmazione object-oriented con subtyping e polimorfismo.

*Conoscenze.* Conoscenza dei principi della programmazione a oggetti.

*Capacità.* Capacità di costruire programmi a oggetti in modo sistematico, di documentarli e di provarne le proprietà.

*Comportamenti.* Saper costruire applicazioni anche complesse sviluppando nuovi moduli, estendendo moduli esistenti ed integrandoli.

### **Indicazioni metodologiche**

Il corso prevede lezioni ed esercitazioni che mettano in pratica i principi esposti.

### **Programma**

- Interpretazione, compilazione, supporto a run-time.
- Oggetti e astrazione procedurale in Java (richiami).
- Semantica operativa di un nucleo orientato ad oggetti con metodi, classi ed ereditarietà.
- Metodologie per lo sviluppo di programmi sequenziali in Java
  - Eccezioni.
  - Astrazioni sui dati
  - Astrazioni sul controllo (iterazione)
  - Gerarchie di tipi
  - Astrazioni polimorfe.
- Testing e debugging.

Ore lezione: 32 Ore esercitazione: 16

### **Bibliografia**

*Libro di testo:* B.Liskov and J. Guttag, Program Development in Java. Abstraction, Specification and Object-Oriented Design, Addison-Wesley, 2001.

Lucidi del corso reperibili sulla pagina web. Testi e correzioni dei compiti e degli esercizi di esame reperibili sulla pagina web.

### **Modalità di esame**

Scritto e orale.

Ulteriore pagina web del corso: <http://www.di.unipi.it/~maggiolo/lucidi.html>

---

## **Modelli della fisica**

Codice: BB002 Crediti: 6 Semestre: 1 Sigla: MF

Settore disciplinare: FIS/02 - Fisica Teorica,  
Modelli e Metodi Matematici

### **Docente**

Fulvio Cornolti  
*cornolti@df.unipi.it*

### **Prerequisiti**

Anlisi 1, Matematica discreta.

### **Obiettivi di apprendimento**

Sviluppare la capacità di descrivere sistemi complessi con modelli deterministici semplici, anche con la implementazione di simulazioni numeriche.

## Descrizione

Partendo da conoscenze elementari di matematica e fisica si studiano diversi fenomeni "naturalisti" (evoluzione di sistemi meccanici, elettrici, ecologici, economici, termodinamici) e si discutono le possibilità e i limiti di una loro descrizione mediante equazioni differenziali ordinarie. Della poca teoria matematica necessaria e non nota da Analisi I si danno i risultati in forma direttamente utilizzabile per la discussione dei sistemi studiati. Simulazioni numeriche vengono utilizzate per studiare i sistemi complessi.

## Programma

Sistemi deterministici lineari del primo e secondo ordine: esempi fisici-proprietà delle soluzioni-operatore di evoluzione-autovalori e autovettori dell'operatore di evoluzione e loro significato fisico- spazio delle fasi-classificazione degli equilibri-cenni ai sistemi di ordine superiore. Cenni di teoria della risposta lineare, funzione di trasferimento per l'oscillatore armonico-esempi e applicazioni. Sistemi con ritardo: un esempio. Alcuni sistemi non lineari-vari esempi -equilibri multipli, stabilità lineare-cicli di isteresi-cicli limite.

## Bibliografia

in corso di preparazione le dispense

## Modalità di esame

Progetto e orale

---

## Ottimizzazione Combinatoria e reti

Codice: AA413   Crediti: 6   Semestre: 1   Sigla: OCR

Settore disciplinare:            MAT/09 - Ricerca Operativa

### Docente

Maria Grazia Scutellà  
*scut@di.unipi.it*  
*www.di.unipi.it/~scut*  
*stanza n. 363, dip.to di Informatica*  
*tel: 0502212771*

### Prerequisiti

Ricerca Operativa

### Obiettivi di apprendimento

Obiettivo del corso è presentare agli studenti le principali problematiche algoritmiche che nascono nella gestione e nel progetto di reti di comunicazione.

## Descrizione

Vengono presentate metodologie risolutive sia per taluni problemi "di base" dell'Ottimizzazione su Reti, non sviluppate nel corso di Ricerca Operativa, sia per problemi "difficili" di Ottimizzazione Combinatoria e Reti. Le metodologie descritte verranno esemplificate nell'ambito di problemi di progetto e gestione di reti di comunicazione.

## Programma

### 1. Introduzione (4 ore):

- Reti di comunicazione e l'Ottimizzazione Combinatoria.
- Problemi polinomiali e problemi NP-ardui; Ottimizzazione Combinatoria, Programmazione Lineare a numeri Interi e Programmazione Lineare Mista.
- Poliedro, inviluppo convesso, disuguaglianze valide.

### 2. Problemi base di ottimizzazione su rete (8 ore):

- Algoritmo preflow-push per il problema del flusso massimo.
- Il problema di flusso di costo minimo.
- Algoritmo basato su cammini minimi successivi per il problema di flusso di costo minimo.
- Algoritmo basato su cancellazione di cicli per il problema di flusso di costo minimo.

### 3. Problemi di instradamento e progetto ottimo su reti (6 ore):

- Problemi di instradamento (routing), progetto di reti (network design) e progetto+instradamento (design + routing).
- Problemi di localizzazione.
- Algoritmo distribuito per il problema dei cammini minimi.
- Cammini minimi bicriterio: costo minimo e numero minimo di hop.

### 4. Approcci euristici (6 ore):

- Algoritmi greedy: formalizzazione e esemplificazioni.
- Valutazione delle prestazioni: algoritmi approssimati, errore relativo;
- La ricerca locale: formalizzazione ed esemplificazioni;
- Un esempio di metaeuristica: il "simulated annealing".

### 5. Tecniche di rilassamento (10 ore):

- Il rilassamento continuo: formalizzazione ed esemplificazioni.
- Utilizzo dell'informazione primale e duale.
- Il rilassamento Lagrangiano: formalizzazione, proprietà ed esemplificazioni.

- o Metodo del subgradiente.
  - o Euristiche Lagrangiane.
  - o Rilassamenti per eliminazione e surrogati.
6. Algoritmi esatti per problemi NP-ardui (10 ore):
- o Tecnica del Branch and Bound per l'enumerazione implicita.
  - o Strategie di visita, tecniche di separazione, cenni implementativi, esemplificazioni.
  - o Piani di taglio nell'enumerazione implicita.
  - o Programmazione dinamica per cammini bicriterio.

Note: Le ore comprendono lezioni e esercitazioni

## Bibliografia

[Appunti di Ottimizzazione Combinatoria](#)

[Appunti di Ricerca Operativa](#) (Capitolo 2)

L. Wolsey "**Integer Programming**" *Wiley-Interscience*, 1998

R.K. Ahuja, T. L. Magnanti, J. B. Orlin "**Network flows. Algorithms and applications**" *Prentice-Hall*, 1993

## Modalità di esame

Scritto ed orale

Ulteriore pagina web del corso:

<http://www.di.unipi.it/optimize/Courses/courses.html>

---

## Programmazione avanzata

Codice: AA033   Crediti: 12   Semestre: 1   Sigla: PA

Settore disciplinare:            INF/01 - Informatica

## Docente

Giuseppe Attardi  
*attardi@di.unipi.it*  
*www.di.unipi.it/~attardi*  
*stanza n.292, dip.to di Informatica*  
*tel:0502212744*

## Prerequisiti

conoscenza ed esperienza di programmazione in un linguaggio a oggetti (Java, C++)

### **Obiettivi di apprendimento**

L'evoluzione dei sistemi software verso il Web Computing coinvolge una varietà di tecniche e strumenti articolati e complessi, dalla programmazione a oggetti, alla programmazione a componenti, alla programmazione di Web Services. Per poter sfruttare a pieno le potenzialità di queste tecniche, occorre avere una comprensione approfondita della loro struttura e del loro funzionamento.

### **Descrizione**

Il corso presenta i modelli di esecuzione, le tecniche e gli strumenti di programmazione avanzati. Si pone particolare attenzione al supporto necessario per il funzionamento di questi strumenti, in modo che lo studente abbia una visione chiara delle conseguenze che le scelte di design possono comportare a tutti i livelli del sistema. Il corso è tra i fondamentali della Laurea Specialistica in Tecnologie Informatiche.

### **Programma**

#### **Rassegna concetti di base di LP**

1. Syntax (Regular Expressions, Context Free Grammars)
2. Scope
3. Parameter Passing
4. Stack, Heap
5. Runtime
6. Type Systems (System F, Kind)

#### **RunTime**

1. Memory Management
2. Execution Engine
3. Verification
4. Interoperability
5. Link/Load
6. Thread
7. PAL
8. Inheritance, Polymorphism, Virtual Functions, Overloading
9. JIT
10. Delegates, Closures

#### **Generic Programming**

1. C++ templates
2. Java Pizza
3. Generics C#

#### **Class Libraries and Frameworks**

1. .NET Framework Class Library
2. Java Class Library

## **Generative Programming**

1. MetaProgramming
2. Reflection
3. Template C++
4. Aspect Oriented Programming, Subject Oriented Programming
5. Intentional Programming
6. Generators

## **Interoperability**

1. socket
2. linguaggio: CORBA/IDL
3. oggetti: COM+

## **Componenti**

1. COM
2. JavaBeans
3. .NET (Assembly, Reflection, Interfaces, Attributes)

## **Web Services**

- XML, XML-Schema
- SOAP, RPC, Rest
- WSDL
- UDDI

## **Transaction Server (EJB, MTS)**

## **Scripting**

1. Perl, Python, Jscript, PHP, Guile, Rebol

## **Bibliografia**

- Programming Language Pragmatics, Michael L. Scott, Morgan-Kaufmann, 2000.
- Shared Source CLI Essentials, David Stutz, Geoff Shilling, Ted Neward, O'Reilly, 2003.
- Generative Programming: Methods, Tools, and Applications, Krzysztof Czarnecki, Ulrich Eisenecker, Addison-Wesley, 2000.
- Applied Microsoft .NET Framework Programming, Jeffrey Richter, Microsoft Press, 2002.

## **Modalità di esame**

Prova intermedia, Term paper finale e prova orale

Ulteriore pagina web del corso: <http://medialab.di.unipi.it/web/AP/>

---

## Reti di calcolatori A

Codice: AA019   Crediti: 6   Semestre: 1   Sigla: RC

Settore disciplinare:            INF/01 - Informatica

### Docente

Antonio Brogi  
*brogi@di.unipi.it*  
*www.di.unipi.it/~brogi*  
*stanza n. 375, dip.to di Informatica*  
*tel: 0502212790*

### Prerequisiti

Materiale didattico: Dispense e fotocopie distribuite dal docente. Il software utilizzato per laboratorio/esercitazioni è il pacchetto STATA.

### Obiettivi di apprendimento

comprendere le problematiche relative alla realizzazione e all'utilizzo di reti di calcolatori acquisendo familiarità con applicazioni e protocolli relativi al modello TCP/IP.

*Conoscenze.* Essere in grado di comprendere il funzionamento delle reti e in particolare delle applicazioni di rete.

*Capacità.* Essere in grado di progettare e realizzare applicazioni di rete e di prendere decisioni riguardanti organizzazione e gestione di una rete di calcolatori.

*Comportamenti.* Essere consapevoli dell'esistenza di problematiche di sicurezza e privacy legate all'uso delle reti di calcolatori.

### Programma

#### *Introduzione alle reti*

- Servizi (orientati alla connessione e non)
- Protocolli, applicazioni, host e router
- Commutazione di circuito e commutazione di pacchetto
- Reti di accesso e mezzi fisici
- Ritardi e perdite nelle reti
- Livelli protocollari

## *Il livello application*

- Caratteristiche generali del livello application
- World Wide Web (HTTP, cenno a HTML)
- Trasferimento file (FTP)
- Email (SMTP, MIME, POP3/IMAP)
- DNS
- Distribuzione di contenuti (Web caching, P2P)

## *Il livello transport*

- Principi dei servizi di livello transport
- Multiplexing/demultiplexing
- Trasferimento affidabile dei dati
- protocolli stop&wait
- protocolli pipelined (GBN,SR)
- Controllo del flusso
- Controllo della congestione
- Istanziamento e implementazione in Internet
- Protocollo UDP
- Protocollo TCP

## *Il livello network*

- Caratteristiche generali del livello network (servizi offerti)
  - Indirizzi e protocollo IP, ICMP e IPv6
  - Principi di routing
  - Algoritmi link-state, distance vector, routing gerarchico
  - Routing in Internet
  - Routing broadcast e multicast

### *Il livello link*

- Caratteristiche generali del livello link (servizi offerti, adattatori)
  - Protocolli di accesso multiplo (MAC)
  - Indirizzi LAN e ARP
  - Ethernet

### *Sicurezza nelle reti*

- Principi di crittografia
- Protocolli per l'autenticazione
- Integrità dei messaggi (firme digitali e digests)
- Attacchi e contromisure (mapping, sniffing, spoofing, DOS)

### *Introduzione ai servizi Web*

Ore lezione: 30 Ore esercitazione: 18

## **Bibliografia**

J. Kurose & K. Ross. *Computer Networking: A top-down approach featuring the Internet*. [third edition] Addison Wesley (del quale esiste una traduzione in

italiano *Reti di calcolatori e Internet*. Pearson - Addison Wesley [terza edizione]).

### Modalità di esame

L'esame consiste di una prova scritta e di una prova orale. Per accedere alla prova orale è necessario aver superato la prova scritta. Sono previste due prove scritte in itinere; il superamento di entrambe le prove permette di accedere direttamente alla prova orale conclusiva (solo per il primo appello). Durante le prove non è consentito l'uso di libri o appunti.

Ulteriore pagina web del corso:

<http://www.di.unipi.it/~brogi/AttivitaDidattica/Informatica/Reti/>

---

## Reti di calcolatori B

Codice: AA019   Crediti: 6   Semestre: 1   Sigla: RC

Settore disciplinare:            INF/01 - Informatica

### Docente

Gianluigi Ferrari  
*giangi@di.unipi.it*  
*www.di.unipi.it/~giangi*  
*stanza n.358, dip.to di Informatica*  
*tel:0502212766*

### Prerequisiti

Lo studente al termine del corso acquisirà le conoscenze dei meccanismi di base di base che regolano la progettazione di reti di calcolatori: protocolli di comunicazione. Inoltre acquisirà le conoscenze per utilizzare al meglio i servizi offerti dalle reti di calcolatori nel progetto di applicazioni di rete.

### Obiettivi di apprendimento

L'obiettivo del corso è introdurre lo studente alle problematiche relative alla realizzazione ed all'utilizzo di reti di calcolatori illustrando applicazioni e protocolli relativi al modello TCP/IP e all'ambiente Internet.

*Conoscenze.* Lo studente al termine del corso acquisirà le conoscenze dei meccanismi di base di base che regolano la progettazione di reti di calcolatori: protocolli di comunicazione. Inoltre acquisirà le conoscenze per utilizzare al meglio i servizi offerti dalle reti di calcolatori nel progetto di applicazioni di rete.

*Capacità.* Lo studente al termine del corso è in grado di dominare l'insieme di tecnologie alla base dello sviluppo delle reti di calcolatori ed il calcolo distribuito

*Comportamenti.* Lo studente al termine del corso ha acquisito una livello di

autonomia che permette di valutare le caratteristiche delle reti di elaboratori e le tecnologie necessarie per il loro progetto e realizzazione. Sarà inoltre cosciente delle problematiche inerenti allo sviluppo di servizi di rete innovativi in contesti operativi.

### **Programma**

- Introduzione alle reti di calcolatori [4 ore]
- Livello application (HTTP, FTP, SMTP, DNS) [8 ore]
- Livello Transport (UDP, TCP) [8 ore]
- Livello Network (IP, Routing) [8 ore]
- Protocolli MAC e standard LAN [4 ore]
- Middleware di rete (RMI, CORBA) [4 ore]
- Sicurezza (cenni) [4 ore]

### **Bibliografia**

*J. Kurose, D. Ross, Internet e Reti di Calcolatori* Terza Edizione (Pearson Ed.). 2005

L. Peterson, B. Davies *Computer Networks: A System Approach* Morgan - Kaufmann, 2000

G. Coulouris, J. Dollimore, T. Kindberg, *Distributed Systems: Concepts and Design*, Addison Wesley (Terza Edizione) 2001

Dispense e note didattiche distribuite dal docente.

### **Modalità di esame**

Valutazioni in itinere, prova scritta e prova orale.

**Ulteriore pagina web del corso:**

<http://www.di.unipi.it/~marcod/Didattica/Reti/index.html>

---

## **Ricerca operativa A**

Codice: AA014   Crediti: 6   Semestre: 2   Sigla: RO

Settore disciplinare:                      MAT/09 - Ricerca Operativa

## Docente

Giancarlo Bigi  
*bigig@di.unipi.it*  
*www.di.unipi.it/~bigig*  
*stanza n. 323, dip.to di Informatica*  
*tel: 0502213124*

## Prerequisiti

La conoscenza dei contenuti dei corsi AA001 "Analisi Matematica", AA005 "Algebra" e AA006 "Algoritmica" è altamente raccomandata.

## Obiettivi di apprendimento

Il corso presenta gli strumenti necessari alla costruzione e alla risoluzione di modelli analitici di ottimizzazione per problemi reali, tipicamente di gestione, di allocazione delle risorse e di logistica. Verranno illustrate le proprietà teoriche ed alcune delle principali tecniche algoritmiche per la soluzione di due grandi classi di problemi di ottimizzazione: problemi di flusso su reti e problemi di programmazione lineare.

*Conoscenze.* Lo studente acquisirà conoscenze sulle principali tecniche di modellazione mediante strumenti logico/matematici, sulle proprietà teoriche dei principali problemi di ottimizzazione su rete e dei problemi programmazione lineare e sui relativi algoritmi risolutivi

*Capacità.* Lo studente saprà analizzare problemi reali, individuandone le caratteristiche principali per fornire una sua modellazione mediante strumenti logico/matematici; studiare e risolvere problemi di ottimizzazione su rete e lineare.

*Comportamenti.* Lo studente acquisirà uno spirito critico nell'analisi di problemi decisionali, nella loro modellazione logico/matematica e nell'analisi dei risultati.

## Descrizione

Il corso si propone di fornire allo studente gli strumenti di base per: analizzare problemi reali, tipicamente di gestione, di allocazione delle risorse e di logistica, individuandone le caratteristiche principali per fornire una sua modellazione mediante strumenti logico/matematici; studiare le proprietà di alcuni problemi di ottimizzazione su rete e descrivere ed utilizzare alcuni algoritmi risolutivi; conoscere le proprietà teoriche della programmazione lineare, derivando da esse gli schemi algoritmici del simplesso primale e del simplesso duale.

## Programma

### *Introduzione (2 ore)*

- Problemi decisionali, problemi di ottimizzazione ed esistenza;
- Classi di problemi; alcuni esempi.

### *Modelli e loro formulazione (12 ore)*

- Tipi di variabili: logiche, continue, discrete;
- Formulazione di vincoli;
- Formulazione della funzione obiettivo.

### *Grafi e reti di flusso (16 ore)*

- Modello generale dei problemi di flusso;
- Alberi, cammini e tagli, visite di grafi e alberi;
- Il problema dei cammini minimi;
- Il problema dell'albero di copertura di costo minimo;
- Il problema del flusso massimo.

### *Programmazione Lineare (18 ore)*

- Modelli di Programmazione Lineare, coppie di problemi duali e teorema debole della dualità
- Vincoli attivi e non attivi, direzioni ammissibili e/o di crescita, Lemma di Farkas;
- Teorema forte della dualità, scarti complementari e interpretazione economica della dualità
- Basi complementari, considerazioni geometriche; algoritmi del Simplexso Primale e del Simplexso Duale;
- Analisi post-ottimale.

Ore lezione: 32 Ore esercitazione: 16

### **Bibliografia**

#### *Testi.*

G..Bigi, A. Frangioni, G. Gallo, S. Pallottino, M.G. Scutellà "Appunti di Ricerca Operativa", *Servizio Editoriale Universitario*

[www.di.unipi.it/optimize/courses/RO/0506/Appunti.html](http://www.di.unipi.it/optimize/courses/RO/0506/Appunti.html)

#### *Esercizi.*

[www.di.unipi.it/optimize/courses/RO/Esercizi/Indice.html](http://www.di.unipi.it/optimize/courses/RO/Esercizi/Indice.html)

[www.di.unipi.it/optimize/courses/RO/SolEsa.html](http://www.di.unipi.it/optimize/courses/RO/SolEsa.html) (soluzioni dettagliate dei compiti d'esame)

### **Modalità di esame**

La valutazione avverrà mediante una prova scritta ed un prova orale. Il superamento della prova scritta costituisce un prerequisito per accedere a quella orale. Il corso prevede due prove di verifica intermedia per l'esonoro dalla prova scritta.

Ulteriore pagina web del corso: <http://www.di.unipi.it/optimize/courses>

Codice: AA014 Crediti: 6 Semestre: 2 Sigla: RO

Settore disciplinare: MAT/09 - Ricerca Operativa

### Docente

Maria Grazia Scutellà  
*scut@di.unipi.it*  
*www.di.unipi.it/~scut*  
*stanza n. 363, dip. to di Informatica*  
*tel: 0502212771*

### Prerequisiti

La conoscenza dei contenuti dei corsi AA001 "Analisi Matematica", AA005 "Algebra" e AA006 "Algoritmica" è altamente raccomandata.

### Obiettivi di apprendimento

Il corso presenta gli strumenti necessari alla costruzione e alla risoluzione di modelli analitici di ottimizzazione per problemi reali, tipicamente di gestione, di allocazione delle risorse e di logistica. Verranno illustrate le proprietà teoriche ed alcune delle principali tecniche algoritmiche per la soluzione di due grandi classi di problemi di ottimizzazione: problemi di flusso su reti e problemi di programmazione lineare.

*Conoscenze.* Lo studente acquisirà conoscenze sulle principali tecniche di modellazione mediante strumenti logico- matematici, sulle proprietà teoriche dei principali problemi di ottimizzazione su rete e dei problemi programmazione lineare e sui relativi algoritmi risolutivi.

*Capacità.* Lo studente saprà analizzare problemi reali, individuandone le caratteristiche principali per fornire una sua modellazione mediante strumenti logico/matematici; studiare e risolvere problemi di ottimizzazione su rete e lineare

*Comportamenti.* Lo studente acquisirà uno spirito critico nell'analisi di problemi decisionali, nella loro modellazione logico-matematica e nell'analisi dei risultati.

### Descrizione

Il corso si propone di fornire allo studente gli strumenti di base per:

- analizzare problemi reali, tipicamente di gestione, di allocazione delle risorse e di logistica, individuandone le caratteristiche principali per fornire una sua modellazione mediante strumenti logico/matematici;
- studiare le proprietà di alcuni problemi di ottimizzazione su rete e descrivere ed utilizzare alcuni algoritmi risolutivi;
- conoscere le proprietà teoriche della programmazione lineare, derivando da esse gli schemi algoritmici del simpleso primale e del simpleso duale.

### Programma

- Introduzione (2 ore) Problemi decisionali, problemi di ottimizzazione ed

esistenza; Classi di problemi; alcuni esempi.

- Modelli e loro formulazione (10 ore) Tipi di variabili: logiche, continue, discrete; Formulazione di vincoli; Formulazione della funzione obiettivo.
- Grafi e reti di flusso (16 ore) Modello generale dei problemi di flusso; Alberi, cammini e tagli, visite di grafi e alberi; Il problema dei cammini minimi; Il problema dell'albero di copertura di costo minimo; Il problema del flusso massimo.
- Programmazione Lineare (20 ore) Modelli di Programmazione Lineare, coppie di problemi duali e teorema debole della dualità Vincoli attivi e non attivi, direzioni ammissibili e/o di crescita, Lemma di Farkas; Teorema forte della dualità, scarti complementari e interpretazione economica della dualità Basi complementari, considerazioni geometriche; algoritmi del Simplex Primale e del Simplex Duale; Analisi post-ottimale.

Ore lezione: 32 Ore esercitazione: 16

## Bibliografia

*Testi.*

G..Bigi, A. Frangioni, G. Gallo, S. Pallottino, M.G. Scutellà "Appunti di Ricerca Operativa", Servizio Editoriale Universitario  
[www.di.unipi.it/optimize/courses/RO/0405/Appunti.html](http://www.di.unipi.it/optimize/courses/RO/0405/Appunti.html)

*Esercizi.*

[www.di.unipi.it/optimize/courses/RO/Esercizi/Indice.html](http://www.di.unipi.it/optimize/courses/RO/Esercizi/Indice.html)

[www.di.unipi.it/optimize/courses/RO/SolEsa.html](http://www.di.unipi.it/optimize/courses/RO/SolEsa.html) (soluzioni dettagliate dei compiti d'esame)

## Modalità di esame

La valutazione avverrà mediante una prova scritta ed un prova orale. Il superamento della prova scritta costituisce un prerequisito per accedere a quella orale. Il corso prevede due prove di verifica intermedia per l'esonero dalla prova scritta.

Ulteriore pagina web del corso: <http://www.di.unipi.it/optimize/courses/>

---

## AA034 Sicurezza nelle reti informatiche (CFU:12)

mutuato da:

### Sicurezza nelle reti informatiche 1

(Corso di Laurea Specialistica in Informatica)

Codice: AA286 Crediti: 6 Semestre: 2 Sigla: SR1

Settore disciplinare: INF/01 - Informatica

#### Docente

Fabrizio Baiardi  
*baiardi@di.unipi.it*  
stanza n. 368, dip.to di Informatica  
tel: 0502212762

#### Programma

- Analisi dei rischi: individuazione di vulnerabilità, minacce ed attacchi
- Valutazione dei rischi conseguenti
- Determinazione delle possibili contromisure
- Valutazione dei costi e del ritorno dell'investimento
- Metodologie open source per l'analisi del rischio

## Sicurezza nelle reti informatiche 2

*(Corso di Laurea Specialistica in Informatica)*

Codice: AA287 Crediti: 6 Semestre: 2 Sigla: SR2

Settore disciplinare: INF/01 - Informatica

#### Docente

Fabrizio Baiardi  
*baiardi@di.unipi.it*  
stanza n. 368, dip.to di Informatica  
tel: 0502212762

#### Programma

- Studio di strumenti (hardware e software) da utilizzare per proteggere il sistema e permettere agli utenti di interagire con esso in maniera sicura
- Firewall ed ids
- Approcci a signature e statistici
- Metodologie e strumenti informatici necessari per
- a) riportare il sistema ad un corretto funzionamento
- b) individuare i dati che sono stati violati
- c) identificare gli attaccanti

---

## Simulazione

Codice: AA042   Crediti: 6   Semestre: 2   Sigla: SIM

Settore disciplinare:            MAT/09 - Ricerca Operativa

### Docente

Giorgio Gallo  
*gallo@di.unipi.it*  
*www.di.unipi.it/~gallo*  
*stanza n.286, dip.to di Informatica*  
*tel:0502212714*

### Obiettivi di apprendimento

Il corso si propone di fornire allo/a studente/ssa gli strumenti di base per la costruzione e l'utilizzo di modelli di simulazione. Un rilevante spazio verrà dato alle applicazioni. Gli argomenti trattati verranno sviluppati anche attraverso piccoli progetti da svolgere durante l'anno: tali progetti costituiranno una componente essenziale nella valutazione conclusiva.

Dopo una introduzione alla simulazione, verranno presentate le principali tecniche di simulazione, con particolare attenzione alla simulazione discreta. Una parte consistente del corso sarà dedicata alle metodologie statistiche per l'analisi dei dati di input e di output. Nell'ultima parte del corso verranno presentati i principali concetti della System Dynamics. Piccoli progetti comportanti attività di modellazione, programmazione ed analisi statistica di dati verranno assegnati durante lo svolgimento del corso.

### Descrizione

Dopo una introduzione alla simulazione, verranno presentate le principali tecniche di simulazione, con particolare attenzione alla simulazione discreta. Una parte consistente del corso sarà dedicata alle metodologie statistiche per l'analisi dei dati di input e di output. Nell'ultima parte del corso verranno presentati i principali concetti della System Dynamics. Piccoli progetti comportanti attività di modellazione, programmazione ed analisi statistica di dati verranno assegnati durante lo svolgimento del corso.

### Programma

1. Introduzione ai modelli di simulazione
2. Simulazione discreta
  - o Il sistema da modellare
  - o Approcci alla modellazione

- Simulazione per processi e per attività
3. Funzioni di distribuzione e test statistici
    - Variabili casuali
    - Distribuzioni discrete e continue
    - Stima di parametri
    - Test di ipotesi
  4. Analisi e scelta dei dati di input
    - Distribuzioni empiriche
    - Analisi dei dati di input
    - Numeri pseudocasuali
  5. Analisi dei dati di output
    - Analisi del transitorio
    - Tecniche per la riduzione della varianza
  6. Modelli di Dinamica dei Sistemi
    - Diagrammi causali
    - Livelli e flussi
    - Ritardi
    - Exponential smoothing
    - Scelta dell'unità di tempo e metodi di integrazione

## Bibliografia

### Testi di riferimento

**A.M. Law & W.D. Kelton** *Simulation Modeling & Analysis*, McGraw-Hill, 1991

**A.M. Mood, F.A. Graybill and D.C. Boes** *Introduction to the theory of statistics*, McGraw-Hill, 1974

**M. Pidd** *Computer Simulation in Management Science*, J.Wiley, 1992

**S.M. Ross** *A course in Simulation*, Macmillan, 1990

### Modalità di esame

Scritto e orale

Ulteriore pagina web del corso:

<http://www.di.unipi.it/di/groups/optimize/Courses/courses.html>

---

# Sistemi Informativi Territoriali

Codice: AA048   Crediti: 6   Semestre: 1   Sigla: SIT

Settore disciplinare:            INF/01 - Informatica

## Docente

Paolo Mogorovich  
*p.mogorovich@cpr.it*  
*www.di.unipi.it/~mogorov*  
*tel: 0502212700*

## Prerequisiti

Nozioni base di matematica e di geometria.

## Obiettivi di apprendimento

Fornire le conoscenze di base sulla rappresentazione dell'informazione geografica e sulla sua gestione all'interno di un'Organizzazione

## Descrizione

Vengono presentati all'inizio i concetti base della rappresentazione cartografica e della cartografia numerica. Quindi si descrivono i modelli per la rappresentazione dello spazio e gli strumenti concettuali e tecnici per la rappresentazione di entità territoriali. Si presentano poi le funzionalità fondamentali degli strumenti GIS: il processo di acquisizione, correlato alle diverse tipologie di fonti di dati, le principali funzioni elaborative, con particolare attenzione all'operazione di incrocio, e infine le tecniche di restituzione. L'attività di laboratorio permetterà di prendere dimestichezza con uno dei più diffusi Software GIS commerciali e di verificare su di esso i principi esposti nelle lezioni teoriche. L'attività di laboratorio avrà luogo in un laboratorio specificamente attrezzato e riservato agli studenti del corso, localizzato nell'area di ricerca del CNR di San Cataldo.

## Programma

In aula: Elementi di cartografia e cartografia numerica. Le caratteristiche dell'informazione geografica. Strumenti di rappresentazione dell'informazione geografica. Un Sistema Informativo Territoriale all'interno di un Ente. Fonti, strumenti e metodi di acquisizione di dati geografici. Query spaziali. L'operazione di incrocio. Metodi e strumenti per la restituzione di dati territoriali. In laboratorio: Introduzione ad ArcView 8.1: Architettura del sistema. ArcCatalog: Gestione dei Dati. Collegamenti a banche dati esterne. Documentazione dei Dati. Import, export di dati. Ricerca dei dati. ArcMap: Creazione ed edizione di mappe. Lavoro con le tabelle di attributi. Interrogazioni. Spatial Analyst: L'analisi spaziale: cammini minimi, interpolazione di rasters, analisi di superfici, gradienti, esposizione,

ombreggiature, punti di osservazione. ArcToolBox: trasformazioni dei dati, batch processing.

## **Bibliografia**

Burrough P.A. - Principles of Geographical Information Systems for Land Resources Assessment, Clarendon Press, Oxford 1989. D.J. Maguire, M. F. Goodchild and D. W. Rhind, Geographical Information Systems, Longman Scientific & Technical, New York, 1991 M.F.Worboys, GIS: a Computer Perspective, Taylor and Francis, London, 1995 M.N.DeMers, Fundamentals of Geographic Information Systems, J.Wiley&Sons (1997) R.Laurini,D.Thompson, Fundamentals of Spatial Information Systems, Academic Press (1992)

## **Modalità di esame**

Scritto e orale

---

## **Sistemi informativi territoriali: laboratorio**

Codice: AA049 Crediti: 6 Semestre: 2 Sigla: SIL

Settore disciplinare: INF/01 - Informatica

### **Docente**

Ernesto Lastres  
*lastres@di.unipi.it*  
*tel: 0502212700*

### **Prerequisiti**

La programmazione. Nozioni di base sui Sistemi Informativi Territoriali. La progettazione concettuale e logica delle basi di dati.

### **Obiettivi di apprendimento**

Illustrare il ciclo di vita e la progettazione di un sistema informativo territoriale. Permettere allo studente di acquisire competenze ed esperienza sul funzionamento di strumenti commerciali per la gestione di SIT e sulla progettazione e realizzazione di SIT.

### **Descrizione**

Vengono presentate le fasi del ciclo di vita di un Sistema Informativo Territoriale (SIT). Viene approfondita in particolare la fase di progettazione concettuale e logica di un SIT. Vengono poi presentate, con un forte supporto di attività di laboratorio, funzionalità avanzate dei sistemi di gestione SIT e la realizzazione di applicazioni attraverso la programmazione. Viene infine svolto

un case study. L'attività di laboratorio avrà luogo in un laboratorio specificamente attrezzato e riservato agli studenti del corso, localizzato presso il Dipartimento di Scienze della Terra in Via Santa Maria oppure nell'area di ricerca del CNR di San Cataldo.

### **Programma**

In aula: Il ciclo di vita di un Sistema Informativo Territoriale (SIT). La progettazione concettuale e logica di un SIT. In laboratorio: Approfondimento della tecnologia GIS: utilizzo di funzioni avanzate, programmazione del SIT. Sviluppo di un case-study di SIT.

### **Modalità di esame**

Esame scritto e orale sulla progettazione del SIT. Valutazione del lavoro in laboratorio e del lavoro sul Case Study.

---

## **Sistemi Intelligenti I**

Codice: AA035   Crediti: 6   Semestre: 1   Sigla: SI1

Settore disciplinare:            INF/01 - Informatica

### **Docente**

Maria Simi  
*simi@di.unipi.it*  
*www.di.unipi.it/~simi*  
*stanza n. 365, dip.to di Informatica*  
*tel: 0502212758*

### **Prerequisiti**

Linguaggio e metodi della matematica, Fondamenti di programmazione, Algoritmica

### **Obiettivi di apprendimento**

Introduzione al settore dell'Intelligenza Artificiale.

*Conoscenze.* Acquisire una conoscenza di base dei concetti, delle tecniche e dei settori applicativi dell'Intelligenza Artificiale.

*Capacità.* Capacità di impostare e risolvere un problema secondo il paradigma di ricerca in uno spazio di stati, capacità di rappresentazione di conoscenza nei principali linguaggi (logici e non) e di progettazione e sviluppo di sistemi di inferenza.

*Comportamenti.* Consapevolezza dei limiti attuali e delle potenzialità dei sistemi

"intelligenti".

## Descrizione

Il corso introdurrà l'Intelligenza Artificiale mediante una trattazione delle tecniche per la realizzazione di agenti intelligenti: risoluzione dei problemi come ricerca euristica in spazi di stati, rappresentazione della conoscenza e ragionamento, pianificazione e apprendimento automatico.

## Programma

Introduzione (2 ore)

- Obiettivi e caratterizzazione dell'Intelligenza Artificiale.
- La visione dell'Intelligenza Artificiale come costruzione di "agenti intelligenti".

Risoluzione dei problemi come ricerca euristica (10 ore + 4 ore di esercitazione)

- Formulazione dei problemi come ricerca in uno spazio di stati.
- Algoritmi di ricerca non informata ed euristica.
- Problemi di soddisfacimento di vincoli.
- Giochi con avversario.

Rappresentazione della conoscenza e ragionamento (12 ore + 10 ore esercitazione)

- Introduzione alla rappresentazione della conoscenza: la mediazione tra espressività e complessità.
- Il calcolo proposizionale e la soddisfacibilità
- Il calcolo dei predicati come linguaggio di rappresentazione: uso e limitazioni.
- Deduzione automatica: il metodo di risoluzione e relative strategie di risoluzione.
- Introduzione alla programmazione logica.
- Sistemi a regole.
- Rappresentazioni strutturate: reti semantiche, rappresentazioni a "frame"
- Logiche descrittive e Web semantico.

Pianificazione (2 ore + 2 ore di esercitazione)

- Il calcolo di situazioni e il problema del contorno.
- Pianificazione nello spazio delle situazioni.
- Pianificazione nello spazio dei piani.

Ore lezione: 30

Ore esercitazione: 18

Ore laboratorio: 0

Ore  
seminari: 0

## Bibliografia

Lucidi delle lezioni.

S. Russell, P. Norvig, "Intelligenza Artificiale: un approccio moderno", Prentice Hall, 2005 (II edizione).

### Modalità di esame

La valutazione dello studente avverrà secondo una delle seguenti modalità:

- Prove in itinere (compitini): entrambe le prove devono essere sufficienti; il voto risultante potrà essere verbalizzato durante uno degli appelli di gennaio e febbraio come voto finale dell'esame.
- Gli appelli sono di norma "solo scritti". Durante gli appelli di gennaio e febbraio è consentito recuperare anche solo uno dei due compitini o migliorarne il risultato.

Ulteriore pagina web del corso: <http://www.di.unipi.it/~simi/AI/SI2005/>

---

## Sistemi Intelligenti II

Codice: AA036   Crediti: 6   Semestre: 2   Sigla: SI2

Settore disciplinare:      INF/01 - Informatica

### Docente

Antonina Starita  
*starita@di.unipi.it*  
*www.di.unipi.it/~starita*  
*stanza n.295, dip.to di Informatica*  
*tel:0502212715*

### Modalità di esame

Scritto e orale

---

## Sistemi Operativi A

Codice: AA015   Crediti: 6   Semestre: 2   Sigla: SO

Settore disciplinare:      INF/01 - Informatica

### Docente

Piero Maestrini  
*maestrin@di.unipi.it*

*www.di.unipi.it/~maestrin*  
*stanza n. 366, dip.to di Informatica*  
*tel: 0502212736*

## **Prerequisiti**

La frequenza dei corsi: Algoritmica, Architetture degli elaboratori, Laboratorio di programmazione concorrente è raccomandata

## **Obiettivi di apprendimento**

Il corso introduce i principi ed i concetti su cui si basano i sistemi operativi e analizza la loro realizzazione nei sistemi reali. In particolare, vengono presentate le tecniche che consentono di coordinare e gestire le risorse di un sistema di elaborazione e che permettono di trasformare la macchina fisica in una macchina astratta, dotata di funzionalità più convenienti per l'utente. Il corso lascia ampio spazio alla descrizione e all'esemplificazione di come i vari concetti presentati sono realizzati nei sistemi Unix/Linux e in quelli della famiglia Microsoft Windows.

*Conoscenze.* Lo studente raggiungerà la conoscenza delle problematiche dei sistemi operativi e delle caratteristiche dei sistemi attualmente più usati

*Capacità.* Lo studente saprà progettare un sistema operativo e saprà valutare la tecnologia dei sistemi operativi commerciali sulla base della piena comprensione delle soluzioni tecniche utilizzate.

*Comportamenti.* Lo studente saprà essere indipendente nella valutazione delle tecnologie e degli strumenti forniti dai sistemi operativi. Sarà inoltre cosciente delle problematiche relative alla loro realizzazione.

## **Programma**

### **Introduzione (2 Ore)**

- Il sistema operativo: ruolo, funzionalità e struttura;
- Evoluzione dei sistemi operativi: batch, multiprogrammazione, time-sharing
- Richiami sul funzionamento dell'elaboratore: interruzioni e loro gestione, I/O,
- Stato del processore, chiamate di sistema.

### **Organizzazione dei Sistemi Operativi (2 ore)**

- Funzionalità
- Struttura: sistemi monolitici e modulari; sistemi stratificati; microkernel
- Organizzazione e funzionalità dei sistemi operativi Unix e Windows.

### **I processi (6 ore)**

- Il concetto di processo
- Stati dei processi
- Realizzazione dei processi nel Sistema Operativo

- Operazioni sui processi
- Processi e thread
- La gestione dei processi e dei thread in Unix e in Windows: stati, rappresentazione, gestione (scheduling), operazioni e comandi

### **Interazione tra processi (6 ore)**

- Interazione mediante memoria condivisa: sincronizzazione; sezioni critiche e mutua esclusione; i semafori; strumenti hardware per la sincronizzazione: test-and-set
- Interazione mediante scambio di messaggi: comunicazione diretta/indiretta, simmetrica/asimmetrica; buffering.
- Interazione tra processi in Unix: comunicazione mediante pipe e fifo; sincronizzazione mediante segnali
- Interazione tra processi in Windows

### **Gestione del processore (4 ore)**

- Concetti generali e politiche di scheduling
- Algoritmi di scheduling: FCFS, SJF, con priorità, Round Robin, con code multiple.
- Scheduling in Unix e in Windows.

### **Gestione della Memoria (8 ore)**

- Concetti generali: spazio degli indirizzi logico/fisico; collegamento
- Allocazione della memoria: contigua: a partizione singola e partizioni multiple; frammentazione;
- Allocazione della memoria non contigua: paginazione e segmentazione
- Memoria virtuale
- Gestione della memoria in Unix e in Windows.

### **Gestione dei dispositivi di I/O (4 ore)**

- I dispositivi di I/O e la loro gestione: concetti generali
- Organizzazione e gestione dei dischi
- Dischi RAID

### **Gestione degli Archivi (6 ore)**

- File system e sua realizzazione
- Struttura logica; metodi di accesso
- Struttura fisica; allocazione dei file
- Protezione
- Il file system di Unix: organizzazione logica e fisica, comandi e system calls per la gestione e l'accesso a file e directories
- Il file system di Windows

### **Cenni sui sistemi operativi di rete (2 ore)**

- Modello OSI e rete Internet

- Modello cliente-servente
- Esecuzione remota di comandi
- Il caso di Unix: socket

Ore lezione: 24 Ore esercitazione: 24

## Bibliografia

### Libro di testo

- Paolo Ancilotti, Aurelio Boari, Anna Ciampolini, Giuseppe Lipari: Sistemi operativi, Mc Graw-Hill, 2004.

### *Altri Testi Consigliati:*

- Andrew S. Tanenbaum. Modern Operating Systems: Second Edition. Prentice Hall
- Silberschatz, P. Galvin: Sistemi Operativi (5<sup>a</sup> edizione), Addison Wesley, 1998.
- P. Maestrini, Sistemi Operativi, Mc Graw-Hill, 1994.
- Uresh Vahalia. Unix Internals: the new frontiers. PrenticeHall 1995.
- Dave Solomon and Mark Russinovic. Inside Microsoft Windows 2000, 3rd Edition MICP ISBN: 0-7356-1021-5.

### Modalità di esame

Scritto e orale Prova scritta sostituibile con due prove scritte intermedie.

Ulteriore pagina web del corso: <http://www.di.unipi.it/~maestrin/>

---

## Sistemi Operativi B

Codice: AA015 Crediti: 6 Semestre: 2 Sigla: SO

Settore disciplinare: INF/01 - Informatica

### Docente

Stefano Chessa  
*ste@di.unipi.it*  
*www.di.unipi.it/~ste*  
*stanza n.321, dip.to di Informatica*  
*tel:0502213122*

### Prerequisiti

La frequenza dei corsi: Algoritmica, Architetture degli elaboratori, Laboratorio di programmazione concorrente è raccomandata

## Obiettivi di apprendimento

Il corso introduce i principi ed i concetti su cui si basano i sistemi operativi e analizza la loro realizzazione nei sistemi reali. In particolare, vengono presentate le tecniche che consentono di coordinare e gestire le risorse di un sistema di elaborazione e che permettono di trasformare la macchina fisica in una macchina astratta, dotata di funzionalità più convenienti per l'utente. Il corso lascia ampio spazio alla descrizione e all'esemplificazione di come i vari concetti presentati sono realizzati nei sistemi Unix/Linux e in quelli della famiglia Microsoft Windows.

*Conoscenze.* Lo studente raggiungerà la conoscenza delle problematiche dei sistemi operativi e delle caratteristiche dei sistemi attualmente più usati.

*Capacità.* Lo studente saprà progettare un sistema operativo e saprà valutare la tecnologia dei sistemi operativi commerciali sulla base della piena comprensione delle soluzioni tecniche utilizzate.

*Comportamenti.* Lo studente saprà essere indipendente nella valutazione delle tecnologie e degli strumenti forniti dai sistemi operativi. Sarà inoltre cosciente delle problematiche relative alla loro realizzazione.

## Programma

### Introduzione (2 Ore)

- Il sistema operativo: ruolo, funzionalità e struttura;
- Evoluzione dei sistemi operativi: batch, multiprogrammazione, time-sharing
- Richiami sul funzionamento dell'elaboratore: interruzioni e loro gestione, I/O,
- Stato del processore, chiamate di sistema.

### Organizzazione dei Sistemi Operativi (2 ore)

- Funzionalità
- Struttura: sistemi monolitici e modulari; sistemi stratificati; microkernel
- Organizzazione e funzionalità dei sistemi operativi Unix e Windows.

### I processi (6 ore)

- Il concetto di processo
- Stati dei processi
- Realizzazione dei processi nel Sistema Operativo
- Operazioni sui processi
- Processi e thread
- La gestione dei processi e dei thread in Unix e in Windows: stati, rappresentazione, gestione (scheduling), operazioni e comandi

### Interazione tra processi (6 ore)

- Interazione mediante memoria condivisa: sincronizzazione; sezioni critiche e mutua esclusione; i semafori; strumenti hardware per la

sincronizzazione: test-and-set

- Interazione mediante scambio di messaggi: comunicazione diretta/indiretta, simmetrica/asimmetrica; buffering.
- Interazione tra processi in Unix: comunicazione mediante pipe e fifo; sincronizzazione mediante segnali
- Interazione tra processi in Windows

### **Gestione del processore (4 ore)**

- Concetti generali e politiche di scheduling
- Algoritmi di scheduling: FCFS, SJF, con priorità, Round Robin, con code multiple.
- Scheduling in Unix e in Windows.

### **Gestione della Memoria (8 ore)**

- Concetti generali: spazio degli indirizzi logico/fisico; collegamento
- Allocazione della memoria: contigua: a partizione singola e partizioni multiple; frammentazione;
- Allocazione della memoria non contigua: paginazione e segmentazione
- Memoria virtuale
- Gestione della memoria in Unix e in Windows.

### **Gestione dei dispositivi di I/O (4 ore)**

- I dispositivi di I/O e la loro gestione: concetti generali
- Organizzazione e gestione dei dischi
- Dischi RAID

### **Gestione degli Archivi (6 ore)**

- File system e sua realizzazione
- Struttura logica; metodi di accesso
- Struttura fisica; allocazione dei file
- Protezione
- Il file system di Unix: organizzazione logica e fisica, comandi e system calls per la gestione e l'accesso a file e directories
- Il file system di Windows

### **Cenni sui sistemi operativi di rete (2 ore)**

- Modello OSI e rete Internet
- Modello cliente-server
- Esecuzione remota di comandi
- Il caso di Unix: socket

Ore lezione: 24 Ore esercitazione: 24

### **Bibliografia**

### **Libro di testo**

- Paolo Ancilotti, Aurelio Boari, Anna Ciampolini, Giuseppe Lipari: Sistemi operativi, Mc Graw-Hill, 2004.

*Altri Testi Consigliati:*

- Andrew S. Tanenbaum. Modern Operating Systems: Second Edition. Prentice Hall
- Silberschatz, P. Galvin: Sistemi Operativi (5<sup>a</sup> edizione), Addison Wesley, 1998.
- P. Maestrini, Sistemi Operativi, Mc Graw-Hill, 1994.
- Uresh Vahalia. Unix Internals: the new frontiers. PrenticeHall 1995.
- Dave Solomon and Mark Russinovic. Inside Microsoft Windows 2000, 3rd Edition MICP ISBN: 0-7356-1021-5.

**Modalità di esame**

Scritto e orale Prova scritta sostituibile con due prove scritte intermedie.

Ulteriore pagina web del corso: <http://www.di.unipi.it/~ste/SO/SO.htm>

---

## Storia delle teorie dell'Informatica

Codice: AA187   Crediti: 3   Semestre: 2   Sigla: STO

Settore disciplinare:      INF/01 - Informatica

**Docente**

Giorgio Germano  
*germano@di.unipi.it*  
stanza n.296a, dip.to di Informatica  
tel:0502212731

---

## Storia ed Applicazioni dell'elaborazione del linguaggio naturale

Codice: AA340   Crediti: 3   Semestre: 2   Sigla: SAN

Settore disciplinare:      INF/01 - Informatica

**Docente**

Amedeo Cappelli

*cappelli@di.unipi.it*  
*www.di.unipi.it/~cappelli*  
*tel: 0502212700*

## **Obiettivi di apprendimento**

Presentare alcuni problemi dell'elaborazione del linguaggio naturale nel loro sviluppo storico.

## **Descrizione**

Verranno presentate alcune motivazioni teoriche e applicative del trattamento automatico del linguaggio naturale. Saranno trattati alcuni specifici problemi anche in riferimento al loro sviluppo storico.

## **Programma**

Elaborazione del linguaggio naturale: aspetti teorici e motivazioni applicative. La traduzione automatica Il text processing I sistemi per la comprensione delle lingue scritte e orali L'elaborazione delle lingue nella società dell'informazione: l'elaborazione del contenuto dei testi, recupero multilingue di informazione, estrazione di informazione

## **Bibliografia**

Cappelli A., Le tecnologie del linguaggio umano nella società dell' informazione: aspetti, problemi e applicazioni dell'elaborazione del contenuto, (dattiloscritto).  
Cappelli A., Strumenti per l'elaborazione lessicale, AIIA Notizie, Anno XIV, 3 (2001), pp. 15-21. Castelfranchi J., Stock O., Macchine come noi, Bari: Laterza, 2000. Russel S. J., Norvig P., Intelligenza Artificiale: un approccio moderno, Torino: UTET, 1998.

## **Modalità di esame**

orale

Ulteriore pagina web del corso: <http://www.di.unipi.it/~cappelli/>

---

## **Tecniche di specifica e dimostrazione**

Codice: AA031   Crediti: 6   Semestre: 2   Sigla: TSD

Settore disciplinare:      INF/01 - Informatica

## **Docente**

Ugo Montanari

*ugo@di.unipi.it*  
*www.di.unipi.it/~ugo*  
*stanza n.280, dip.to di Informatica*  
*tel: 0502212721*

## **Prerequisiti**

Elementi di Programmazione  
Elementi di Logica

## **Obiettivi di apprendimento**

La definizione formale di semplici linguaggi di programmazione.

## **Descrizione**

Il corso presenta i principi della semantica operativa, i principi della semantica denotazionale e le tecniche per confrontarle nel caso di un linguaggio imperativo e di un linguaggio funzionale di ordine superiore. Viene anche presentata la semantica operativa e la semantica osservazionale di un linguaggio per la descrizione di processi.

## **Programma**

- Introduzione e sistemi di prova con regole di inferenza. (2h)
- Sintassi e semantica operativa di un semplice linguaggio imperativo (IMP). (4h)
- Domini e costruzioni di domini. (4h)
- Teorema del minimo punto fisso. (2h)
- Segnature e algebre dei termini. (2h)
- Semantica denotazionale di IMP. (2h)
- Tecniche di prova per induzione. (4h)
- Equivalenza tra semantica operativa e denotazionale di IMP. (4h)
- Sintassi e semantica operativa lazy di un linguaggio funzionale higher order (HOL). (4h)
- Semantica denotazionale lazy di HOL. (2h)
- Relazione tra semantica operativa e denotazionale di HOL. (2h)
- Sintassi e semantica operativa di un linguaggio per la descrizione di processi (CCS). (4h)
- Semantica osservazionale del CCS. (4h)

Ore lezione: 30 Ore esercitazione: 10

## **Bibliografia**

- Glynn Winskel, "The formal Semantics of Programming Languages", MIT Press, 1993. Capitoli: 1.3, 2, 3, 4, 5, 8, 11.
- Robin Milner, "Communication and Concurrency", Prentice Hall, 1989. Capitoli: 1-7, 10.

## **Modalità di esame**

Scritto e orale

Ulteriore pagina web del corso: <http://www.di.unipi.it/~ugo/TSD.html>

---

## Tecnologia dell'informazione e della comunicazione nelle organizzazioni

Codice: AA020 Crediti: 3 Semestre: 1 Sigla: TIC

Settore disciplinare: INF/01 - Informatica

### Docente

Giovanni A. Cignoni  
*giovanni@di.unipi.it*  
*www.di.unipi.it/~giovanni*  
*stanza n. 320, dip.to di Informatica*  
*tel: 0502213128*

### Prerequisiti

Il corso non ha prerequisiti obbligatori. Tuttavia, una certa familiarità con le funzionalità dei sistemi informativi (Basi di dati) e le problematiche dello sviluppo di software (Ingegneria del software) è utile per una matura comprensione dei temi affrontati dal corso.

### Obiettivi di apprendimento

Il corso si propone di dare allo studente la capacità di comprendere il contesto e le tendenze delle applicazioni delle TIC nelle organizzazioni pubbliche e private.

### Descrizione

La pianificazione e l'uso delle TI sono strettamente connesse ai processi gestionali e alla comprensione delle necessità di un'organizzazione. Oltre le competenze tecniche, la consapevolezza dei vantaggi e delle conseguenze dell'applicazione delle TI è necessaria all'informatico per non correre il rischio di proporre le sue soluzioni prima di conoscere e comprendere i problemi.

I temi trattati dal corso coprono il syllabus dell'area *pianificazione* del livello base della certificazione [EUCIP](#).

### Programma

I temi affrontati dal corso sono:

- le organizzazioni e le TI;

- la gestione delle TI;
- l'economia dell'informatica;
- la nuova economia;
- pianificazione e controllo dei progetti;
- comunicazione e presentazione;
- aspetti etici e legali.

## Bibliografia

Le copie dei lucidi usati in aula sono disponibili [in linea](#).

Testi di lettura e approfondimento sono:

- G. Bracchi, C. Francalanci, G. Motta, "Sistemi informativi per l'impresa digitale", McGraw-Hill, 2005;  
Testo per l'approfondimento dei temi relativi ai sistemi informativi aziendali.
- P.F. Camussone, "Informatica, organizzazione e strategie", McGraw-Hill, 2000;  
Testo di approfondimento sull'impatto delle TI nelle organizzazioni.
- P. Schgör, R. Brambilla, F. Amarilli, "Professione informatica - Vol. 1 Pianificazione, uso e gestione dei sistemi informativi", Franco Angeli, 2004.  
Testo organizzato secondo il syllabus EUCIP, livello base, area pianificazione.

## Modalità di esame

I 3 crediti formativi associati al corso corrispondono, secondo la formula standard, a un carico di studio di 75 ore complessive. Le condizioni per il riconoscimento dei crediti (vedi anche [Regolamento per i corsi di cultura di contesto](#)) sono:

- la presenza ad almeno l'80% delle ore di lezione in aula (21 ore);
- il superamento di due prove in itinere.

Le prove in itinere sono organizzate come test in parte a risposta multipla e in parte a risposta aperta. La parte a risposta multipla segue lo stile delle prove per la certificazione EUCIP.

Durante le prove non è consentito consultare materiale didattico o appunti. Le prove si svolgeranno durante il corso nei previsti periodi di sospensione della didattica.

Modalità alternative per il riconoscimento dei crediti (seminario o relazione) possono essere concordate con il docente all'inizio del corso (ad esempio per studenti lavoratori per i quali la frequenza è un problema).

Ulteriore pagina web del corso: <http://www.di.unipi.it/~giovanni/>

---

# Tecnologie dell'informazione: mercato, società e cultura

Codice: AA414 Crediti: 3 Semestre: 1 Sigla: TMS

Settore disciplinare: INF/01 - Informatica

## Docente

Ippolito Spadafora  
*ippos@di.unipi.it*  
*stanza n.355, dip.to di Informatica*  
*tel:0502212788*

## Obiettivi di apprendimento

Tramite l'IT (Information Technology), milioni di persone sono collegate e correlate in nuovi spazi che vanno modificando la nostra più profonda identità, il modo in cui pensiamo, la forma e l'organizzazione delle nostre comunità. È importante quindi capire i profondi cambiamenti che le nuove tecnologie dell'informazione stanno producendo, anche se le conseguenze non sono facilmente prevedibili. In particolare è importante conoscere le caratteristiche del contesto socio-economico di oggi per cercare di intravedere cosa ci riserva il futuro; studiare i nuovi fenomeni in corso è importante per poter capire il loro effetto sulla società, sulla cultura, sull'economia.

## Programma

1. L'Età dell'Informazione
2. Le problematiche dell'IT
3. L'uomo e l'IT
4. Il senso del sé e l'IT
5. Le aziende e l'IT
6. Il mercato dell'informazione
7. La Globalizzazione
8. La Teoria della Fiducia
9. Il commercio elettronico

## Bibliografia

Spadafora I., Tecnologie dell'Informazione: mercato, società, cultura, Boringhieri, Torino, 2003  
Spadafora I., Filosofia del computer: aspetti e implicazioni, Pubblicazione Interna, TR-00-07 del Dipartimento di Informatica, Università di Pisa, Giugno 2000

## Modalità di esame

Test (da definire)

---

## Tecnologie Informatiche per l'Impresa

Codice: AA186 Crediti: 3 Semestre: 2 Sigla: TII

Settore disciplinare: INF/01 - Informatica

### Docente

Pietro Piram  
*piram@di.unipi.it*  
stanza n.374, dip.to di Informatica  
tel: 0502212750

### Programma

- **azienda e processi aziendali**
- **tecnologie di analisi**
- **l'hardware** le cpu, le unità periferiche, le interconnessioni,
- **il software di base** sistemi operativi, ambienti di sviluppo, data base, software di rete
- **architetture e sistemi** internet, la distribuzione delle applicazioni, ERP e applicazioni satellite
- **il software applicativo** Gli ERP, Componenti orizzontali, Componenti verticali (industry solutions), Le applicazioni satellite.

### Modalità di esame

Scritto e orale

Ulteriore pagina web del corso: <http://www.cli.di.unipi.it/~piram/>

---

## Teoria della programmazione

Codice: AA050 Crediti: 6 Semestre: 1 Sigla: TP

Settore disciplinare: INF/01 - Informatica

### Docente

Giorgio Germano  
*germano@di.unipi.it*  
stanza n.296a, dip.to di Informatica  
tel: 0502212731

## Prerequisiti

Cultura matematica di base. Qualche conoscenza di programmazione ricorsiva.

## Obiettivi di apprendimento

Questa lezione intende fornire alcune tecniche matematiche elementari da applicare alle procedure ricorsive per 1. scriverne più pulitamente il testo, 2. capirne più compiutamente il funzionamento, 3. determinarne più agevolmente le proprietà .

## Descrizione

Impareremo anzitutto a vedere la nota programmazione deterministica come un caso speciale di programmazione indeterministica, facendo uso di parallelismo concorrente e non concorrente. Nell' ambito più ampio della programmazione indeterministica sarà molto facile vedere come la ricorsività è connessa con l'induttività. A questo punto si ottengono immediatamente le applicazioni interessanti per le procedure ricorsive usuali. Speciale attenzione sarà dedicata alla modularità dei procedimenti.

## Programma

Programmazione deterministica e indeterministica. Sequenzialità, parallelismo, coniugazione. Insiemi induttivi, relazioni ricorsive e funzioni ricorsive. Algebre di Peano.

## Bibliografia

Testo: dispense fornite dal docente. Riferimenti: R. Dedekind, Was sind und was sollen die Zahlen ? , Vieweg 1888 ; G. Birkhoff, Lattice Theory, Amer. Math. Soc. Colloquium 1967 ; B. A. Davey et alii , Introduction to Lattices and order , Cambridge University Press, 1990 ; S. Mac Lane et alii , Sheaves in Geometry and Logic, Springer 1992 .

## Modalità di esame

Progetto ed esame orale.

---

## Teoria dell'informazione

Codice: AA041   Crediti: 6   Semestre: 2   Sigla: TI

Settore disciplinare:      INF/01 - Informatica

Docente

Francesco Romani  
*romani@di.unipi.it*  
*www.di.unipi.it/~romani*  
*stanza n.326, dip.to di Informatica*  
*tel: 0502212734*

## **Prerequisiti**

Calcolo delle probabilità, calcolo numerico

## **Obiettivi di apprendimento**

Obiettivo del corso è introdurre i concetti essenziali della teoria classica della Informazione e Trasmissione.

*Conoscenze.* Conoscenza dei principi della Teoria dell'Informazione.

*Capacità.* Capacità di svolgere semplici esercizi di Teoria dell'Informazione.

*Comportamenti.* La conoscenza dei fondamenti della teoria dell'Informazione non dovrebbe avere effetti sui comportamenti degli allievi.

## **Descrizione**

Il corso introduce i concetti essenziali della teoria dell'informazione secondo Shannon: entropia, sorgenti, canali, codifica della sorgente e del canale, teoremi di codifica in assenza di rumore e in presenza di rumore. Vengono poi trattati i codici correttori a blocchi ed in maggiore dettaglio i codici ciclici. Viene anche data un'introduzione alla teoria della comunicazione nel modello continuo.

## **Indicazioni metodologiche**

Il processo di apprendimento è organizzato in lezioni teoriche su argomenti in sequenza logica ed esercitazioni .

## **Programma**

1. Concetti Generali di Teoria dell'Informazione. Modelli di sistemi per la trasmissione dati. Modello discreto e modello continuo. Problemi e risultati.
2. La funzione entropia. Definizione. Proprietà. Entropia condizionata e informazione reciproca. Un'applicazione: la proprietà di equiripartizione asintotica.
3. Sorgenti d'informazione discreta. Definizione. Tipi di sorgenti. Sorgente estensione e sorgente adiacente.
4. Codifica in assenza di rumore (codifica della sorgente). Definizione di codice. Tipi di codice. Diseguaglianza di Kraft e MacMillan. Lunghezza media. Il teorema della codifica in assenza di rumore. Costruzione dei codici ottimali: il metodo di Huffman.
5. I canali discreti senza memoria. Definizione. Capacità, Schemi di decisione e probabilità di errore. Il canale binario simmetrico. La codifica del canale. Il teorema fondamentale.
6. Codifica in presenza di rumore (codifica del canale). Problematica. Un

- risultato asintotico fondamentale: il teorema di Shannon. L'inverso debole.
7. I codici correttori a blocchi. Distanza di Hamming. I codici lineari. Algoritmi di codifica e decodifica. Capacità correttiva. Probabilità di errore. I codici di Hamming. Validità del teorema fondamentale.
  8. I Codici ciclici. Definizione. Casi binari e non binari. caratterizzazioni mediante matrici e polinomi. I codici BCH. I codici di Reed Solomon.

Ore lezione: 32    Ore esercitazione: 15    Ore laboratorio: 0    Ore seminari: 0

### **Bibliografia**

Piram, Romani "Appunti di Teoria dell'Informazione" Manoscritto disponibile sotto forma di fotocopie

### **Modalità di esame**

Scritto e orale

Ulteriore pagina web del corso: <http://www.cli.di.unipi.it/~romani/TI.html>

---