



SAPIENZA  
UNIVERSITÀ DI ROMA

# Informatica (2024)

## Il corso

Codice corso: 29923

Classe di laurea: L-31

Durata: 3 anni

Lingua: ITA

Modalità di erogazione:

Dipartimento: INFORMATICA

## Presentazione

Il Corso di Laurea in Informatica ha l'obiettivo di formare figure professionali capaci di affrontare con successo le sfide costituite dalle crescenti esigenze della società dell'informazione. Le laureate e i laureati in Informatica acquisiranno una solida preparazione culturale di base, che permetterà loro di mantenersi al passo col progredire delle tecnologie, oltre a una salda preparazione tecnica, che consentirà loro un rapido inserimento professionale nel settore delle tecnologie dell'informazione e della comunicazione. Le laureate e i laureati in Informatica saranno inoltre in grado di accedere ai livelli di studio universitario successivi al primo, sia nel settore dell'informatica che in altri settori scientifici. La preparazione culturale di base permetterà alle laureate e ai laureati in Informatica di acquisire: - familiarità col metodo scientifico di indagine; - capacità di comprendere ed utilizzare strumenti matematici di supporto; - conoscenze metodologiche e competenze di base in un ampio spettro di settori delle scienze e delle tecnologie dell'informazione e della comunicazione; - familiarità con almeno una lingua dell'Unione Europea. A questo scopo, nei primi due anni del Corso, uguali per tutti, sono previsti insegnamenti di base ritenuti indispensabili alla formazione culturale, metodologica e tecnica di una laureata o un laureato in Informatica. Durante il terzo anno, oltre a completare la propria formazione, le studentesse e gli studenti potranno scegliere corsi che ne caratterizzeranno il profilo in senso più marcatamente metodologico o applicativo, seguendo le alternative di completamento proposte. La formazione si completa con un tirocinio, da svolgere internamente, approfondendo tematiche avanzate con la supervisione di un docente, o esternamente presso aziende del settore, tipicamente seguendo attività di analisi, progettazione e sviluppo software. Il Corso di Laurea ha attivato diverse convenzioni Erasmus, che permettono alle studentesse e agli studenti di svolgere esperienze di formazione all'estero. Il Corso di Laurea è a numero programmato e l'immatricolazione è quindi subordinata al superamento di una prova di ammissione. Il corso di laurea è in possesso del Bollino GRIN 2019, rilasciato dalla comunità dei docenti universitari italiani nel settore INF/01, che certifica la qualità della formazione informatica di livello universitario.

# Percorso formativo

## Metodologico

### 1° anno

Insegnamento	Semestre	CFU	Lingua
101226   CALCOLO DIFFERENZIALE	1°	6	ITA

#### Obiettivi formativi

##### Obiettivi generali:

acquisire una conoscenza delle tecniche elementari del Calcolo Differenziale e delle principali applicazioni a problemi di massimo-minimo e allo studio del grafico di funzioni.

##### Obiettivi specifici:

Conoscenza e comprensione: al termine del corso lo studente avrà acquisito le nozioni e i risultati di base del Calcolo Differenziale con particolare attenzione ai concetti di funzione, limite di funzione e derivata.

Applicare conoscenza e comprensione: al termine del corso lo studente sarà in grado di risolvere semplici problemi del Calcolo Differenziale, quali il calcolo esplicito di derivate, il calcolo del massimo e minimo locale e globale di funzioni di una variabile, e il disegno approssimativo del grafico di funzioni.

Capacità critiche e di giudizio: lo studente avrà le basi per utilizzare un grafico come strumento di analisi di una situazione concreta descrivibile matematicamente.

Capacità comunicative: lo studente sarà in grado di comprendere un testo scientifico di complessità non elevata e di riassumerne i concetti principali.

Capacità di apprendimento: le conoscenze acquisite permetteranno uno studio, individuale o impartito in un corso, relativo ad aspetti più avanzati del Calcolo Differenziale e al Calcolo Integrale.

1015883   FONDAMENTI DI PROGRAMMAZIONE	1°	9	ITA
---	----	---	-----

**Insegnamento****Semestre****CFU****Lingua****Obiettivi formativi**

## Obiettivi generali

Introduzione alla programmazione tramite il linguaggio Python.

## Obiettivi specifici

Introduzione alla programmazione tramite il linguaggio Python.

Tipi di dati, variabili, assegnamenti, strutture di controllo, funzioni, classi, moduli e Input/Output.

Strutture dati: vettori, stringhe, liste, tuple e dizionari.

Progettazione e sviluppo di programmi tramite programmazione procedurale e orientata agli oggetti.

Algoritmi ricorsivi ed iterativi.

Librerie di Python per la grafica, per la gestione dei file, per l'elaborazione di testi/html e per l'accesso ad Internet.

Debugging e testing di programmi.

## Conoscenza e comprensione

Comprendere e definire i requisiti di un problema.

Decidere come rappresentare le informazioni in input e quali strutture dati usare per le elaborazioni intermedie e per l'output.

Definire l'algoritmo di soluzione.

Codificare l'algoritmo sotto forma di programma Python.

Modularizzare il programma in piccole funzioni/metodi separate.

Verificare tramite tests che il programma segua i requisiti.

## Applicazione di conoscenza e comprensione

Lo studente dovrà realizzare dei compiti di programmazione per casa, scansionati durante il corso, per mettere in pratica e dimostrare le conoscenze apprese.

Alla fine del corso la prova d'esame sarà basata su una prova in laboratorio in cui lo studente dovrà risolvere e programmare alcuni esercizi.

## Autonomia di giudizio

Lo studente alla fine del corso deve essere in grado di scegliere autonomamente come risolvere un problema di programmazione (analisi, implementazione e test).

## Abilità comunicative

Nella fase di analisi del problema e definizione dei requisiti è importante avere una buona capacità di comprensione del linguaggio.

## Capacità di apprendimento successivo

Le basi dell'analisi di un problema per comprendere le specifiche e progettare sia le strutture dati necessarie che l'algoritmo più adatto è applicabile ad altri linguaggi di programmazione e potrà aiutare nei successivi corsi di programmazione.

1020420 | METODI  
MATEMATICI PER  
L'INFORMATICA

1°

6

ITA

**Insegnamento****Semestre****CFU****Lingua****Obiettivi formativi**

## Obiettivi generali:

L'insegnamento è indirizzato all'acquisizione delle conoscenze logiche e insiemistiche di base per affrontare lo studio di altri argomenti in matematica ed informatica.

## Obiettivi specifici:

## Conoscenza e comprensione:

Alla fine del corso lo studente avrà piena comprensione degli strumenti logico-insiemistici proposti

## Applicare conoscenza e comprensione:

Avrà acquisito la capacità di portare avanti un rigoroso, anche se elementare, ragionamento matematico, in particolare per quanto riguarda i principi logici fondamentali e l'uso dell'induzione in tutte le sue forme.

## Capacità critiche e di giudizio:

Sarà quindi capace di affrontare criticamente gli argomenti proposti in altri insegnamenti sia teorici che applicati. A questo proposito vengono forniti molti esempi presi da altri insegnamenti.

## Capacità di comunicare quanto si è appreso:

La partecipazione attiva in classe e l'uso dell'esame orale servono a stimolare lo studente nell'acquisire il linguaggio proprio della matematica e a trasmettere le conoscenze e le capacità acquisite in maniera appropriata.

Capacità di proseguire lo studio: Lo studente sarà in grado di approfondire nello studio personale i temi trattati usando quanto appreso come base.

1015880 |  
PROGETTAZIONE DI  
SISTEMI DIGITALI

1°

6

ITA

**Obiettivi formativi**

## Obiettivi generali:

Metodologie di progettazione di circuiti combinatori e sequenziali.

## Obiettivi specifici:

codifica binaria di vari tipi di dato, algebra booleana, analisi e sintesi di circuiti combinatori, flip-flop, analisi e sintesi di circuiti sequenziali, registri, interconnessione tra registri e altri moduli

## Conoscenza e comprensione:

Conoscere e capire come l'elaboratore gestisce ed elabora l'informazione

## Applicare conoscenza e comprensione:

Dimostrare di saper progettare semplici circuiti combinatori e sequenziali in grado di svolgere determinati compiti.

## Capacità critiche e di giudizio:

saper scegliere il miglior approccio, tra i vari studiati, per risolvere un determinato compito

## Capacità comunicative:

essere in grado di valutare e motivare le proprie scelte nella progettazione di un circuito

## Capacità di apprendimento:

Capire le differenze e i vantaggi delle varie tecniche di progettazione.

1015881 |  
ARCHITETTURA DEGLI  
ELABORATORI

2°

6

ITA

**Insegnamento****Semestre****CFU****Lingua****Obiettivi formativi**

## Obiettivi generali:

L'obiettivo dell'insegnamento di Architettura degli elaboratori è di far comprendere i principi che sono usati per progettare i calcolatori moderni. In particolare, il corso tratta la struttura interna del microprocessore e le idee che hanno permesso la straordinaria evoluzione della potenza di calcolo negli ultimi 30 anni. come pipelining, caching, branch prediction, e multi-processing.

## Obiettivi specifici:

Il corso tratta i principi di base di organizzazione del microprocessore e le nozioni di pipelining, caching, branch prediction, virtualizzazione e multi-processing. Inoltre, il corso tratta la programmazione assembly.

## Conoscenza e comprensione:

Lo studente acquisirà conoscenza sull'organizzazione del microprocessore MIPS, come implementazione delle idee generali che fanno parte degli obiettivi del corso. Inoltre, lo studente acquisirà conoscenza su come si strutturano i programmi in assembly, incluse le strutture dati, i paradigmi standard di programmazione e la ricorsione.

## Applicazione di conoscenza e comprensione:

Le conoscenze sono applicate sulla architettura MIPS, comprese in modo tale da poter capire le implicazioni delle scelte di programmazione sulla performance dei programmi su hardware specifico. Questo risultato è ottenuto tramite esercizi di programmazione e di valutazione delle prestazioni.

## Autonomia di giudizio:

Lo studente sarà in grado di comprendere le problematiche relative alle prestazioni del software su hardware specifico e di valutarne autonomamente le caratteristiche.

## Abilità comunicative:

Il corso non si propone espliciti obiettivi sulle capacità comunicative, eccetto di formare all'esposizione rigorosa degli argomenti tecnici.

## Capacità di apprendimento successivo:

Il corso pone le basi per la comprensione dei moduli dell'insegnamento di Sistemi operativi e di tutti i corsi di programmazione, inclusa la programmazione di sistemi paralleli.

97796 | CALCOLO  
INTEGRALE

2°

6

ITA

**Obiettivi formativi**

## Obiettivi generali:

Lo scopo del corso è fornire strumenti di analisi matematica di base come integrali, serie numeriche e di potenze ed equazioni differenziali.

## Obiettivi specifici:

## Conoscenza e comprensione:

L'obiettivo è che lo studente apprenda alcune tecniche di integrazione per calcolare integrali definiti ed indefiniti usando la formula di integrazione per parti e l'integrazione per sostituzione.

Lo studente sarà in grado di studiare la convergenza delle serie numeriche e approssimare, in alcuni casi, la loro somma e studierà le serie di potenze e gli sviluppi in serie di Taylor delle funzioni elementari. Infine imparerà a risolvere alcune equazioni differenziali ordinarie del primo ordine a variabili separabili e lineari del primo ordine ed equazioni differenziali del secondo ordine lineari con coefficienti costanti omogenee e non.

## Applicazione di conoscenza e comprensione:

Il corso prevede lezioni teoriche ed esercitazioni per apprendere lo svolgimento degli esercizi pratici.

A meta' del corso è prevista una prova pratica per verificare l'apprendimento del programma svolto.

1015885 |  
INTRODUZIONE AGLI  
ALGORITMI

2°

6

ITA

**Obiettivi formativi**

## Obiettivi generali

Introduzione ai metodi di base per la progettazione e l'analisi degli algoritmi, iterativi e ricorsivi, ed alla valutazione della loro efficienza computazionale.

## Obiettivi specifici

## Conoscenza e comprensione:

Al termine del corso gli studenti conosceranno le metodologie di base per la progettazione e l'analisi di algoritmi iterativi e ricorsivi, le strutture dati elementari, alcuni modi per scandire tali strutture, i principali algoritmi di ordinamento e le implementazioni più elementari dei dizionari.

## Applicare conoscenza e comprensione:

Al termine del corso gli studenti avranno acquisito familiarità con le principali strutture dati di base, in particolare quelle che implementano i dizionari. Sapranno spiegarne gli algoritmi e analizzarne la complessità, evidenziando come le prestazioni dipendano dalla struttura dati utilizzata. Saranno in grado di progettare nuove strutture dati e i relativi algoritmi, rielaborando quelli esistenti; sapranno spiegare i principali algoritmi di ordinamento, illustrando le strategie di progetto sottostanti e le relative analisi di complessità; saranno in grado di confrontare i comportamenti asintotici dei tempi di esecuzione degli algoritmi studiati; saranno in grado di progettare soluzioni ricorsive di problemi e di analizzare asintoticamente gli algoritmi risultanti.

## Capacità critiche e di giudizio:

Lo studente avrà le basi per analizzare la qualità di un algoritmo e delle relative strutture dati, sia dal punto di vista della effettiva risoluzione del problema che da quello della efficienza computazionale con la quale il problema viene risolto.

## Capacità comunicative:

Lo studente acquisirà la capacità di esporre in modo chiaro ed organizzato le proprie conoscenze, capacità che verrà verificata sia mediante i quesiti presentati nelle prove scritte che durante la prova orale. Lo studente sarà in grado di esprimere un'idea algoritmica in modo rigoroso ad alto livello, in pseudocodice.

## Capacità di apprendimento successivo:

Le conoscenze acquisite permetteranno allo studente di affrontare lo studio, individuale o previsto nell'ambito di un corso di laurea magistrale, di tecniche algoritmiche e di strutture dati di base.

**Insegnamento****Semestre****CFU****Lingua****Obiettivi formativi**

## Obiettivi generali:

Apprendimento dei concetti della programmazione orientata agli oggetti mediante il linguaggio di programmazione Java.

## Obiettivi specifici:

I concetti fondamentali della programmazione orientata agli oggetti: classi e oggetti, incapsulamento, ereditarietà, polimorfismo, binding statico e dinamico, i design pattern. La programmazione funzionale. Gli strumenti e le metodologie di base della progettazione software tramite un linguaggio orientato agli oggetti. Il linguaggio Java.

## Conoscenza e comprensione:

Conoscenza dei costrutti dei linguaggi di programmazione orientata agli oggetti, con particolare riferimento al linguaggio Java. Comprensione di un programma Java. Capacità di scrittura di un programma Java di piccole e medie dimensioni.

## Applicare conoscenza e comprensione:

Essere in grado di applicare le metodologie di base per affrontare la progettazione di sistemi software di grandezza medio-piccola. Saper usare i principali strumenti di sviluppo per realizzare tali sistemi in Java.

## Capacità critiche e di giudizio:

Capacità di identificare istruzioni, costrutti o pattern errati o inefficienti così come corretti o efficienti in Java.

## Capacità comunicative:

Illustrazione del progetto sviluppato.

## Capacità di apprendimento successivo:

Capacità di apprendere e applicare nuove tecniche di programmazione a partire da quelle apprese durante il corso.

AAF1101 | LINGUA  
INGLESE

2°

3

ENG

**Obiettivi formativi**

L'insegnamento della lingua inglese punta a mettere in grado lo studente di leggere correttamente la letteratura scientifica in inglese, la documentazione tecnica, e di poter sostenere conversazioni in inglese su aspetti tecnici.

**2° anno****Insegnamento**

1015886 | ALGEBRA

**Semestre**

1°

**CFU**

9

**Lingua**

ITA

**Obiettivi formativi**

## Obiettivi generali:

Fornire conoscenze di base di teoria dei gruppi e di algebra lineare che sono comunemente utilizzati in informatica."

## Obiettivi specifici:

Concetti di Algebra elementari

Introduzione alle strutture algebriche

Sviluppo del linguaggio dell'algebra lineare: spazi vettoriali e loro omomorfismi; spazi vettoriali, numeri e algebra delle matrici; endomorfismi, determinanti e diagonalizzazione; applicazioni.

## Conoscenze e comprensione:

Alla fine del corso lo studente sarà in grado di fare conti in gruppi piccoli di facile struttura e di comprendere il significato e l'utilizzo di matrici nello studio di fenomeni lineari.

## Applicare conoscenza e comprensione:

Utilizzo della diagonalizzazione di operatori per fornire un'interpretazione geometrica ad un fenomeno lineare e ricavarne informazioni rilevanti. Utilizzo del concetto di gruppo nello studio di problemi combinatori finiti.

## Capacità di giudizio:

Gli studenti saranno in grado di scegliere lo strumento algebrico più appropriato allo studio di un problema.

## Capacità di comunicazione:

Gli studenti impareranno il linguaggio dell'algebra lineare e dei gruppi e rudimenti di complessità.

## Capacità di apprendimento:

L'apprendimento del linguaggio lineare e grupppale fornirà agli studenti la capacità di apprendere argomenti che ne fanno uso.

1015887 | BASI DI DATI

1°

12

ITA

**Obiettivi formativi**

## Obiettivi generali:

Esporre gli studenti a solide metodologie di progettazione e realizzazione di applicazioni per basi di dati

## Obiettivi specifici:

Esporre gli studenti a metodologie formali e scalabili per l'analisi concettuale e la progettazione di applicazioni per basi di dati e a tecnologie standard per la loro realizzazione.

## Conoscenza e comprensione:

Gli studenti acquisiranno conoscenze metodologiche fondamentali per la progettazione di applicazioni per basi di dati non banali (in particolare per le fasi di: a) raccolta dei requisiti; b) analisi concettuale dei dati e delle funzionalità; c) progettazione della base dati e delle funzionalità) e per la loro realizzazione (utilizzo di DBMS e di linguaggi standard di definizione, interrogazione e manipolazione dei dati).

## Applicare conoscenza e comprensione:

Gli studenti saranno in grado di applicare in modo efficace le conoscenze indicate al punto precedente in progetti reali di applicazioni per basi di dati non banali.

## Capacità critiche e di giudizio:

Gli studenti saranno in grado di prendere autonomamente decisioni razionali in tutte le fasi del processo di progettazione di applicazioni per basi di dati.

## Capacità comunicative:

Gli studenti saranno in grado di interagire in modo proficuo con i committenti (per quanto concerne la raccolta dei requisiti) e con altri analisti e progettisti (per quanto concerne le attività di analisi e progettazione di sistemi software non banali).

## Capacità di apprendimento:

Gli studenti saranno in grado di ampliare le loro conoscenze in modo autonomo consultando, secondo necessità, manualistica tecnica nell'ambito della progettazione di applicazioni per basi di dati.

I MODULO

1°

6

ITA

**Obiettivi formativi**

## Obiettivi generali:

Esporre gli studenti a solide metodologie di progettazione e realizzazione di applicazioni per basi di dati

## Obiettivi specifici:

Esporre gli studenti a metodologie formali e scalabili per l'analisi concettuale e la progettazione di applicazioni per basi di dati e a tecnologie standard per la loro realizzazione.

## Conoscenza e comprensione:

Gli studenti acquisiranno conoscenze metodologiche fondamentali per la progettazione di applicazioni per basi di dati non banali (in particolare per le fasi di: a) raccolta dei requisiti; b) analisi concettuale dei dati e delle funzionalità; c) progettazione della base dati e delle funzionalità) e per la loro realizzazione (utilizzo di DBMS e di linguaggi standard di definizione, interrogazione e manipolazione dei dati).

## Applicare conoscenza e comprensione:

Gli studenti saranno in grado di applicare in modo efficace le conoscenze indicate al punto precedente in progetti reali di applicazioni per basi di dati non banali.

## Capacità critiche e di giudizio:

Gli studenti saranno in grado di prendere autonomamente decisioni razionali in tutte le fasi del processo di progettazione di applicazioni per basi di dati.

## Capacità comunicative:

Gli studenti saranno in grado di interagire in modo proficuo con i committenti (per quanto concerne la raccolta dei requisiti) e con altri analisti e progettisti (per quanto concerne le attività di analisi e progettazione di sistemi software non banali).

## Capacità di apprendimento:

Gli studenti saranno in grado di ampliare le loro conoscenze in modo autonomo consultando, secondo necessità, manualistica tecnica nell'ambito della progettazione di applicazioni per basi di dati.

**Obiettivi formativi**

## Obiettivi generali:

Acquisire conoscenza e capacità di applicazione di argomenti di base di probabilità e statistica.

## Obiettivi specifici:

Assiomi e proprietà elementari delle probabilità. Variabili Aleatorie. Distribuzioni continue e discrete. Valori attesi. Introduzione all'inferenza statistica.

## Conoscenza e comprensione:

Al termine del corso lo studente avrà acquisito le nozioni e i risultati di base relativi alla teoria della probabilità su spazi finiti e numerabili, al concetto di vettore aleatorio discreto e al concetto di variabile aleatoria continua.

## Applicare conoscenza e comprensione:

Gli studenti che abbiano superato l'esame saranno in grado di applicare le nozioni di calcolo combinatorio per risolvere semplici problemi di probabilità discreta, problemi inerenti vettori casuali discreti e numeri casuali rappresentati da variabili aleatorie continue. Lo studente sarà anche in grado di apprezzare il significato e le implicazioni dell'indipendenza e del condizionamento (nell'ambito di modelli discreti), comprendere il significato di alcuni teoremi limite fondamentali, quali la legge dei grandi numeri.

## Capacità critiche e di giudizio:

Lo studente avrà le basi per analizzare e costruire modelli probabilistici in semplici situazioni di interesse fisico, biologico e tecnologico, utilizzare tavole e software di simulazione delle leggi discrete di più comune applicazione, nonché della legge gaussiana, e di comprendere l'utilizzazione di strumenti statistici elementari nell'inferenza, nel campionamento statistico e nella simulazione.

## Capacità comunicative:

Capacità di esporre i contenuti nella parte orale della verifica e negli eventuali quesiti teorici presenti nella prova scritta.

Capacità di apprendimento: le conoscenze acquisite permetteranno uno studio, individuale o impartito in un corso relativo ad aspetti più specialistici di teoria della probabilità.

**Obiettivi formativi**

## Obiettivi generali:

Il corso ha come obiettivi i concetti, la struttura, e i meccanismi dei sistemi operativi. Verranno trattate caratteristiche fondamentali, presenti fin dai sistemi più tradizionali, ma anche peculiarità dei sistemi moderni che nascono come conseguenza dell'evoluzione ricorrente della tecnologia.

Conoscere ed usare l'interfaccia di programmazione fra sviluppatore software e kernel relativamente ai servizi base di accesso alle risorse del sistema operativo Linux.

## Obiettivi specifici:

Caratteristiche e concetti dei sistemi operativi moderni, con particolare riferimento ai sistemi Unix e Linux. Si inizierà con una descrizione dell'evoluzione dei sistemi operativi nel tempo, per continuare con concetti fondamentali come i processi, lo stallo, e relativi meccanismi di prevenzione, la concorrenza tra processi, la gestione della memoria, processore e I/O, i file system e la sicurezza.

Conoscere i concetti (processi, espressioni regolari, file system) e i comandi shell più importanti di Linux. Saper creare script in Bash. Saper scrivere programmi C che usino le system call di Linux.

## Conoscenza e comprensione:

Capire in modo profondo come i sistemi operativi danno supporto all'esecuzione dei programmi degli utenti e gestiscono le periferiche hardware di un computer. Metodi e tecniche fondamentali per la rappresentazione dei processi in memoria e la gestione efficiente di multiprogrammazione—molteplici processi eseguiti contemporaneamente in un sistema con risorse limitate.

Conoscenza del funzionamento interno del sistema operativo Linux. Conoscenza del funzionamento della shell Bash. Fondamenti del linguaggio C. Conoscenza delle principali system call di Linux.

## Applicare conoscenza e comprensione:

Progettare programmi a livello utente e di sistema in modo efficiente e sicuro.

Saper creare script in Bash in grado di risolvere problemi pratici. Saper scrivere programmi in C che sfruttino le system call di Linux per ottimizzare l'uso delle risorse.

## Capacità critiche e di giudizio:

Essere in grado di predire l'uso delle risorse richieste da un programma, di scoprire una possibile situazione di stallo in un sistema multiprogrammato, garantire la mutua esclusione tra processi e l'accesso protetto a zone di memoria o risorse sensibili.

Saper valutare la soluzione più appropriata per ottenere un determinato risultato, usando o singoli comandi shell, o uno script Bash, o un programma C basato su system call di Linux.

## Capacità comunicative:

Saper comunicare in modo chiaro e preciso le caratteristiche dei sistemi operativi e i loro meccanismi di supporto software/hardware.

Essere in grado di comunicare e documentare script Bash e programmi C basati su system call di Linux.

## Capacità di apprendimento:

Saper sfruttare la conoscenza acquisita nella progettazione di sistemi e di programmi utente nel modulo successivo del corso. Essere in grado di usare questa conoscenza nell'apprendimento di proprietà di sistemi più complessi come quelli distribuiti e cloud.

Essere in grado di usare i concetti appresi in corsi avanzati da sistemista aziendale, o in un eventuale corso avanzato che richieda interazione con Linux, come ad es.: programmazione di sistema, cloud computing, sistemi distribuiti, cybersecurity.

**Obiettivi formativi****Obiettivi generali:**

Il corso ha come obiettivi i concetti, la struttura, e i meccanismi dei sistemi operativi. Verranno trattate caratteristiche fondamentali, presenti fin dai sistemi più tradizionali, ma anche peculiarità dei sistemi moderni che nascono come conseguenza dell'evoluzione ricorrente della tecnologia.

**Obiettivi specifici:**

Il corso coprirà le caratteristiche e concetti dei sistemi operativi moderni, con particolare riferimento ai sistemi Unix e Linux. Si inizierà con una descrizione dell'evoluzione dei sistemi operativi nel tempo, per continuare con concetti fondamentali come i processi, lo stallo, e relativi meccanismi di prevenzione, la concorrenza tra processi, la gestione della memoria, processore e I/O, i file system e la sicurezza.

**Conoscenza e comprensione:**

Capire in modo profondo come i sistemi operativi danno supporto all'esecuzione dei programmi degli utenti e gestiscono le periferiche hardware di un computer. Metodi e tecniche fondamentali per la rappresentazione dei processi in memoria e la gestione efficiente di multiprogrammazione—molteplici processi eseguiti contemporaneamente in un sistema con risorse limitate.

**Applicare conoscenza e comprensione:**

Progettare programmi a livello utente e di sistema in modo efficiente e sicuro.

**Capacità critiche e di giudizio:**

Essere in grado di predire l'uso delle risorse richieste da un programma, di scoprire una possibile situazione di stallo in un sistema multiprogrammato, garantire la mutua esclusione tra processi e l'accesso protetto a zone di memoria o risorse sensibili.

**Capacità comunicative:**

Saper comunicare in modo chiaro e preciso le caratteristiche dei sistemi operativi e i loro meccanismi di supporto software/hardware.

**Capacità di apprendimento:**

Saper sfruttare la conoscenza acquisita nella progettazione di sistemi e di programmi utente nel modulo successivo del corso. Essere in grado di usare questa conoscenza nell'apprendimento di proprietà di sistemi più complessi come quelli distribuiti e cloud.

**Obiettivi formativi**

## Obiettivi generali:

Esporre gli studenti a solide metodologie di progettazione e realizzazione di applicazioni per basi di dati

## Obiettivi specifici:

Esporre gli studenti a metodologie formali e scalabili per l'analisi concettuale e la progettazione di applicazioni per basi di dati e a tecnologie standard per la loro realizzazione.

## Conoscenza e comprensione:

Gli studenti acquisiranno conoscenze metodologiche fondamentali per la progettazione di applicazioni per basi di dati non banali (in particolare per le fasi di: a) raccolta dei requisiti; b) analisi concettuale dei dati e delle funzionalità; c) progettazione della base dati e delle funzionalità) e per la loro realizzazione (utilizzo di DBMS e di linguaggi standard di definizione, interrogazione e manipolazione dei dati).

## Applicare conoscenza e comprensione:

Gli studenti saranno in grado di applicare in modo efficace le conoscenze indicate al punto precedente in progetti reali di applicazioni per basi di dati non banali.

## Capacità critiche e di giudizio:

Gli studenti saranno in grado di prendere autonomamente decisioni razionali in tutte le fasi del processo di progettazione di applicazioni per basi di dati.

## Capacità comunicative:

Gli studenti saranno in grado di interagire in modo proficuo con i committenti (per quanto concerne la raccolta dei requisiti) e con altri analisti e progettisti (per quanto concerne le attività di analisi e progettazione di sistemi software non banali).

## Capacità di apprendimento:

Gli studenti saranno in grado di ampliare le loro conoscenze in modo autonomo consultando, secondo necessità, manualistica tecnica nell'ambito della progettazione di applicazioni per basi di dati.

**Obiettivi formativi**

## Obiettivi generali:

Essere in grado di progettare/valutare le proprietà, la struttura di memorizzazione e i protocolli di un sistema di gestione di basi di dati.

Esporre gli studenti a solide metodologie di progettazione e realizzazione di applicazioni per basi di dati

## Obiettivi specifici:

Conoscenza delle proprietà di uno schema relazionale e di una decomposizione. Capacità di interrogare una base di dati.

Capacità di valutare i costi delle operazioni di accesso ai dati. Conoscenza di protocolli di per il controllo della concorrenza.

Esporre gli studenti a metodologie formali e scalabili per l'analisi concettuale e la progettazione di applicazioni per basi di dati e a tecnologie standard per la realizzazione di basi di dati relazionali.

## Conoscenza e comprensione:

Fondamenti teorici della progettazione e della interrogazione di una base di dati relazionale. Principali strutture di organizzazione dei dati su memoria secondaria. Principali tecniche usate nei DBMS per il controllo dell'esecuzione concorrente delle transazioni.

Gli studenti acquisiranno conoscenze metodologiche fondamentali per la progettazione di applicazioni per basi di dati non banali (in particolare per le fasi di: a) raccolta dei requisiti; b) analisi concettuale dei dati e delle funzionalità; c) progettazione della base dati e delle funzionalità) e per la loro realizzazione (utilizzo di DBMS relazionali e del linguaggio SQL).

## Applicare conoscenza e comprensione:

Progettare schemi relazionali con "buone proprietà". Interrogare una base di dati mediante algebra relazionale. Valutare i costi delle operazioni fondamentali su file con diversi tipi di organizzazione fisica.

Gli studenti saranno in grado di applicare in modo efficace le conoscenze acquisite in progetti reali di applicazioni per basi non banali.

## Capacità critiche e di giudizio:

Essere in grado di valutare le proprietà di uno schema relazionale e di una sua decomposizione. Essere in grado di scegliere la struttura dati più appropriata per memorizzare le informazioni di uno schema.

Gli studenti saranno in grado di prendere autonomamente decisioni razionali in tutte le fasi del processo di progettazione di applicazioni per basi di dati.

## Capacità comunicative:

Essere in grado di comunicare/condividere caratteristiche qualitative/quantitative relative alla struttura relazionale di una base di dati

Gli studenti saranno in grado di interagire in modo proficuo con i committenti (per quanto concerne la raccolta dei requisiti) e con altri analisti e progettisti (per quanto concerne le attività di analisi e progettazione di sistemi software non banali).

## Capacità di apprendimento:

Essere in grado di utilizzare i concetti acquisiti nel modulo successivo del corso e in un eventuale corso avanzato di basi di Dati.

Gli studenti saranno in grado di ampliare le loro conoscenze in modo autonomo consultando, secondo necessità, manualistica tecnica nell'ambito della progettazione di applicazioni per basi di dati.

**Insegnamento****Semestre****CFU****Lingua****Obiettivi formativi**

## Obiettivi generali

Acquisire la conoscenza di base delle più note tecniche algoritmiche di progettazione e delle tecniche di valutazione della correttezza e della complessità degli algoritmi.

## Obiettivi specifici

## Conoscenza e comprensione:

Al termine del corso gli studenti posseggono le conoscenze di base relative a:

- tecniche fondamentali di progettazione algoritmica;
- analisi della correttezza e della efficienza degli algoritmi;

## Applicazione di conoscenza e comprensione:

Al termine del corso gli studenti sono in grado di:

- analizzare le prestazioni di un algoritmo tramite strumenti matematici rigorosi;
- analizzare algoritmi e strutture dati
- progettare ed analizzare nuovi algoritmi, sfruttando le metodologie presentate durante il corso.

## Autonomia di giudizio:

Lo studente alla fine del corso deve essere in grado di scegliere autonomamente qual'è la tecnica algoritmica più adatta da applicare per un determinato problema e valutare tra più soluzioni algoritmiche per un certo problema qual'è da preferirsi.

## Abilità comunicative:

Lo studente acquisirà la capacità di esprimere un'idea algoritmica tramite l'uso di uno pseudocodice.

## Capacità di apprendimento:

Lo studente avrà acquisito la capacità di analizzare un problema, progettare le necessarie strutture dati e un algoritmo corretto ed efficiente che lo risolva.

1015889 | RETI DI  
ELABORATORI

2°

9

ITA

**Obiettivi formativi**

## Obiettivi generali:

Imparare le basi delle reti di elaboratori moderne e Internet.

## Obiettivi specifici:

## Conoscenza e comprensione:

Conoscenza e comprensione dei protocolli dei livelli applicazione, trasporto, rete e collegamento della pila TCP/IP.

## Applicazione di conoscenza e comprensione:

Capacità di comprendere qualsiasi protocollo coinvolto nella comunicazione TCP/IP

Capacità di comprendere strumenti e tecniche per risolvere problemi di rete

Capacità di usare servizi di rete come DNS e DHCP che permettono di far funzionare una rete

## Autonomia di giudizio:

Individuare problemi di rete

Valutare la realizzazione di nuovi servizi

## Abilità comunicative:

Capacità di descrivere le reti in termini strutturali secondo il modello a 5 livelli.

## Capacità di apprendimento successivo:

Il corso fornisce le basi per poter apprendere nozioni di reti wireless e Internet of Things.

1020422 | SISTEMI  
OPERATIVI

2°

12

ITA

**Obiettivi formativi**

## Obiettivi generali:

Il corso ha come obiettivi i concetti, la struttura, e i meccanismi dei sistemi operativi. Verranno trattate caratteristiche fondamentali, presenti fin dai sistemi più tradizionali, ma anche peculiarità dei sistemi moderni che nascono come conseguenza dell'evoluzione ricorrente della tecnologia.

Conoscere ed usare l'interfaccia di programmazione fra sviluppatore software e kernel relativamente ai servizi base di accesso alle risorse del sistema operativo Linux.

## Obiettivi specifici:

Caratteristiche e concetti dei sistemi operativi moderni, con particolare riferimento ai sistemi Unix e Linux. Si inizierà con una descrizione dell'evoluzione dei sistemi operativi nel tempo, per continuare con concetti fondamentali come i processi, lo stallo, e relativi meccanismi di prevenzione, la concorrenza tra processi, la gestione della memoria, processore e I/O, i file system e la sicurezza.

Conoscere i concetti (processi, espressioni regolari, file system) e i comandi shell più importanti di Linux. Saper creare script in Bash. Saper scrivere programmi C che usino le system call di Linux.

## Conoscenza e comprensione:

Capire in modo profondo come i sistemi operativi danno supporto all'esecuzione dei programmi degli utenti e gestiscono le periferiche hardware di un computer. Metodi e tecniche fondamentali per la rappresentazione dei processi in memoria e la gestione efficiente di multiprogrammazione—molteplici processi eseguiti contemporaneamente in un sistema con risorse limitate.

Conoscenza del funzionamento interno del sistema operativo Linux. Conoscenza del funzionamento della shell Bash. Fondamenti del linguaggio C. Conoscenza delle principali system call di Linux.

## Applicare conoscenza e comprensione:

Progettare programmi a livello utente e di sistema in modo efficiente e sicuro.

Saper creare script in Bash in grado di risolvere problemi pratici. Saper scrivere programmi in C che sfruttino le system call di Linux per ottimizzare l'uso delle risorse.

## Capacità critiche e di giudizio:

Essere in grado di predire l'uso delle risorse richieste da un programma, di scoprire una possibile situazione di stallo in un sistema multiprogrammato, garantire la mutua esclusione tra processi e l'accesso protetto a zone di memoria o risorse sensibili.

Saper valutare la soluzione più appropriata per ottenere un determinato risultato, usando o singoli comandi shell, o uno script Bash, o un programma C basato su system call di Linux.

## Capacità comunicative:

Saper comunicare in modo chiaro e preciso le caratteristiche dei sistemi operativi e i loro meccanismi di supporto software/hardware.

Essere in grado di comunicare e documentare script Bash e programmi C basati su system call di Linux.

## Capacità di apprendimento:

Saper sfruttare la conoscenza acquisita nella progettazione di sistemi e di programmi utente nel modulo successivo del corso. Essere in grado di usare questa conoscenza nell'apprendimento di proprietà di sistemi più complessi come quelli distribuiti e cloud.

Essere in grado di usare i concetti appresi in corsi avanzati da sistemista aziendale, o in un eventuale corso avanzato che richieda interazione con Linux, come ad es.: programmazione di sistema, cloud computing, sistemi distribuiti, cybersecurity.

**Insegnamento****Semestre****CFU****Lingua****Obiettivi formativi**

## Obiettivi generali:

Il corso ha come obiettivi i concetti, la struttura, e i meccanismi dei sistemi operativi. Verranno trattate caratteristiche fondamentali, presenti fin dai sistemi più tradizionali, ma anche peculiarità dei sistemi moderni che nascono come conseguenza dell'evoluzione ricorrente della tecnologia.

Conoscere ed usare l'interfaccia di programmazione fra sviluppatore software e kernel relativamente ai servizi base di accesso alle risorse del sistema operativo Linux.

## Obiettivi specifici:

Caratteristiche e concetti dei sistemi operativi moderni, con particolare riferimento ai sistemi Unix e Linux. Si inizierà con una descrizione dell'evoluzione dei sistemi operativi nel tempo, per continuare con concetti fondamentali come i processi, lo stallo, e relativi meccanismi di prevenzione, la concorrenza tra processi, la gestione della memoria, processore e I/O, i file system e la sicurezza.

Conoscere i concetti (processi, espressioni regolari, file system) e i comandi shell più importanti di Linux. Saper creare script in Bash. Saper scrivere programmi C che usino le system call di Linux.

## Conoscenza e comprensione:

Capire in modo profondo come i sistemi operativi danno supporto all'esecuzione dei programmi degli utenti e gestiscono le periferiche hardware di un computer. Metodi e tecniche fondamentali per la rappresentazione dei processi in memoria e la gestione efficiente di multiprogrammazione—molteplici processi eseguiti contemporaneamente in un sistema con risorse limitate.

Conoscenza del funzionamento interno del sistema operativo Linux. Conoscenza del funzionamento della shell Bash.

Fondamenti del linguaggio C. Conoscenza delle principali system call di Linux.

## Applicare conoscenza e comprensione:

Progettare programmi a livello utente e di sistema in modo efficiente e sicuro.

Saper creare script in Bash in grado di risolvere problemi pratici. Saper scrivere programmi in C che sfruttino le system call di Linux per ottimizzare l'uso delle risorse.

## Capacità critiche e di giudizio:

Essere in grado di predire l'uso delle risorse richieste da un programma, di scoprire una possibile situazione di stallo in un sistema multiprogrammato, garantire la mutua esclusione tra processi e l'accesso protetto a zone di memoria o risorse sensibili.

Saper valutare la soluzione più appropriata per ottenere un determinato risultato, usando o singoli comandi shell, o uno script Bash, o un programma C basato su system call di Linux.

## Capacità comunicative:

Saper comunicare in modo chiaro e preciso le caratteristiche dei sistemi operativi e i loro meccanismi di supporto software/hardware.

Essere in grado di comunicare e documentare script Bash e programmi C basati su system call di Linux.

## Capacità di apprendimento:

Saper sfruttare la conoscenza acquisita nella progettazione di sistemi e di programmi utente nel modulo successivo del corso. Essere in grado di usare questa conoscenza nell'apprendimento di proprietà di sistemi più complessi come quelli distribuiti e cloud.

Essere in grado di usare i concetti appresi in corsi avanzati da sistemista aziendale, o in un eventuale corso avanzato che richieda interazione con Linux, come ad es.: programmazione di sistema, cloud computing, sistemi distribuiti, cybersecurity.

**3° anno****Insegnamento****Semestre****CFU****Lingua**

1022301 | INGEGNERIA  
DEL SOFTWARE

1°

6

ITA

**Insegnamento****Semestre****CFU****Lingua****Obiettivi formativi****Obiettivi generali:**

Il corso illustra i fondamenti delle metodologie e degli strumenti per la gestione dei processi software. Particolare attenzione viene dedicata alle metodologie di analisi e progettazione orientate agli oggetti, e alla loro gestione e documentazione mediante UML.

**Obiettivi specifici:**

Introduzione agli approcci all'ingegneria del software e al ciclo di vita del software, approfondimento sulle attività di specifica, analisi, progetto e test di sistemi software, tecniche per la gestione dei processi, con particolare riferimento alla gestione della qualità e dei rischi e all'analisi dei costi.

**Conoscenze e comprensione:**

Alla fine del corso lo studente avrà acquisito conoscenze relative ai principali modelli di ciclo di vita del software, alle metriche per il dimensionamento dello sforzo, alle tecniche di descrizione delle diverse componenti di un progetto software. Saranno state acquisite conoscenze relative all'uso del linguaggio UML. Infine, avranno anche acquisito familiarità con la letteratura scientifica nel campo.

**Applicarei conoscenza e comprensione:**

Alla fine del corso lo studente sarà in grado di lavorare in team alle attività di analisi, progettazione, documentazione e gestione di progetti software di medie dimensioni. Avranno imparato a produrre documentazione basata su UML, relativamente ai principali tipi di diagrammi: dei casi d'uso, delle classi, di interazione, di stato e di attività, anche attraverso l'utilizzo di ambienti software professionali orientati allo sviluppo sistematico di progetti software. Infine saranno in grado di produrre una valutazione dello sforzo basata su Punti Funzione e Punti Use Case.

**Capacità di giudizio:**

Gli studenti svilupperanno le capacità di analisi necessarie per valutare diverse alternative durante il processo di sviluppo software, con particolare riferimento alla valutazione delle scelte architetture e dei rischi di progetto.

**Capacità di comunicazione:**

Gli studenti impareranno a documentare le loro scelte, anche attraverso l'uso di strumenti di generazione della documentazione, in particolare sfruttando notazioni diagrammatiche. Avranno anche acquisito la capacità di preparare presentazioni relative ad argomenti scientifici.

**Capacità di apprendimento successivo:**

La conoscenza degli aspetti di rigore formale alla base della disciplina dell'ingegneria del software permetterà agli studenti di acquisire rapidamente confidenza con tecniche, oltre a quelle considerate nel corso, basate su principi generali.

1041727 | AUTOMI  
CALCOLABILITA' E  
COMPLESSITA'

1°

6

ITA

**Insegnamento****Semestre****CFU****Lingua****Obiettivi formativi**

## Obiettivi generali:

Durante il corso saranno introdotti i più importanti risultati dell'Informatica teorica: a partire dai fondamentali risultati in teoria della calcolabilità degli anni trenta, passando per quelli in teoria degli automi degli anni cinquanta per arrivare al problema aperto P contenuto o uguale a NP, esplicitamente sollevato negli anni settanta.

## Obiettivi specifici:

Gli studenti capiranno che ci sono diversi modelli di computazione e cosa ne determina il potere computazionale.

Gli studenti apprenderanno concetti astratti come classi di linguaggi, macchine universali, riducibilità e sapranno che alcuni problemi non possono essere risolti con un calcolatore e che altri sono computazionalmente difficili da risolvere o addirittura così difficili da poter essere considerati non risolvibili. Faremo vedere come alcuni di questi risultati sono utilizzati oggi.

## Conoscenza e comprensione:

Al termine del corso gli studenti conosceranno i metodi e risultati di base della della teoria degli automi, della calcolabilità e della complessità e sapranno applicarli per individuare la complessità di problemi in diversi campi. In particolare sapranno:

dimostrare l'equivalenza tra le diverse caratterizzazioni dei linguaggi regolari

dimostrare l'equivalenza tra le diverse caratterizzazioni dei linguaggi context-free

spiegare il concetto di non determinismo

giustificare l'esistenza di problemi privi di soluzioni algoritmiche o intrattabili.

## Applicazione di conoscenza e comprensione:

Gli studenti impareranno:

come costruire automi finiti (deterministici e non) da una specifica (formale o informale)

come costruire automi a pila (deterministici e non) da una specifica (formale o informale)

a usare la riducibilità tra problemi per dimostrarne la decidibilità o l'indecidibilità

a usare la riducibilità polinomiale per provare la NP-hardness di un problema

## Autonomia di giudizio:

Capire il giusto livello di astrazione utile per risolvere un problema, scegliere il modello computazionale più conveniente in un determinato contesto applicativo

## Abilità comunicative:

descrivere un linguaggio formale, a parole o attraverso uno degli strumenti offerti di descrizione finita, descrivere problemi indecidibili, intrattabili o trattabili, spiegare il significato e la rilevanza delle classi P ed NP nonché del problema "P=NP?"

## Capacità di apprendimento:

Lo studente sarà in grado di imparare altri modelli computazionali, sia completamente diversi da quelli studiati durante il corso, sia variazioni di questi. Egli sarà capace di capire nuove prove di NP-completezza o più in generale prove di completezza per una qualunque classe di complessità

A SCELTA DELLO  
STUDENTE

1°

6

ITA

A SCELTA DELLO  
STUDENTE

2°

6

ITA

AAF1053 | TIROCINIO

2°

15

ITA

**Obiettivi formativi**

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

AAF1004 | PROVA  
FINALE

2°

6

ITA

**Insegnamento****Semestre****CFU****Lingua****Obiettivi formativi**

La prova finale consiste nella stesura, nella presentazione e nella discussione di una dissertazione scritta, elaborata autonomamente dallo studente, che documenti in modo organico e dettagliato il problema affrontato nell'ambito del tirocinio formativo e tutte le attività compiute per pervenire alla soluzione.

Insegnamenti  
metodologici affini  
Metodologici  
caratterizzanti

Tecnologico**1° anno****Insegnamento****Semestre****CFU****Lingua**

101226 | CALCOLO  
DIFFERENZIALE

1°

6

ITA

**Obiettivi formativi**

## Obiettivi generali:

acquisire una conoscenza delle tecniche elementari del Calcolo Differenziale e delle principali applicazioni a problemi di massimo-minimo e allo studio del grafico di funzioni.

## Obiettivi specifici:

Conoscenza e comprensione: al termine del corso lo studente avrà acquisito le nozioni e i risultati di base del Calcolo Differenziale con particolare attenzione ai concetti di funzione, limite di funzione e derivata.

Applicare conoscenza e comprensione: al termine del corso lo studente sarà in grado di risolvere semplici problemi del Calcolo Differenziale, quali il calcolo esplicito di derivate, il calcolo del massimo e minimo locale e globale di funzioni di una variabile, e il disegno approssimativo del grafico di funzioni.

Capacità critiche e di giudizio: lo studente avrà le basi per utilizzare un grafico come strumento di analisi di una situazione concreta descrivibile matematicamente.

Capacità comunicative: lo studente sarà in grado di comprendere un testo scientifico di complessità non elevata e di riassumerne i concetti principali.

Capacità di apprendimento: le conoscenze acquisite permetteranno uno studio, individuale o impartito in un corso, relativo ad aspetti più avanzati del Calcolo Differenziale e al Calcolo Integrale.

1015883 | FONDAMENTI  
DI PROGRAMMAZIONE

1°

9

ITA

**Insegnamento****Semestre****CFU****Lingua****Obiettivi formativi**

## Obiettivi generali

Introduzione alla programmazione tramite il linguaggio Python.

## Obiettivi specifici

Introduzione alla programmazione tramite il linguaggio Python.

Tipi di dati, variabili, assegnamenti, strutture di controllo, funzioni, classi, moduli e Input/Output.

Strutture dati: vettori, stringhe, liste, tuple e dizionari.

Progettazione e sviluppo di programmi tramite programmazione procedurale e orientata agli oggetti.

Algoritmi ricorsivi ed iterativi.

Librerie di Python per la grafica, per la gestione dei file, per l'elaborazione di testi/html e per l'accesso ad Internet.

Debugging e testing di programmi.

## Conoscenza e comprensione

Comprendere e definire i requisiti di un problema.

Decidere come rappresentare le informazioni in input e quali strutture dati usare per le elaborazioni intermedie e per l'output.

Definire l'algoritmo di soluzione.

Codificare l'algoritmo sotto forma di programma Python.

Modularizzare il programma in piccole funzioni/metodi separate.

Verificare tramite tests che il programma segua i requisiti.

## Applicazione di conoscenza e comprensione

Lo studente dovrà realizzare dei compiti di programmazione per casa, scansionati durante il corso, per mettere in pratica e dimostrare le conoscenze apprese.

Alla fine del corso la prova d'esame sarà basata su una prova in laboratorio in cui lo studente dovrà risolvere e programmare alcuni esercizi.

## Autonomia di giudizio

Lo studente alla fine del corso deve essere in grado di scegliere autonomamente come risolvere un problema di programmazione (analisi, implementazione e test).

## Abilità comunicative

Nella fase di analisi del problema e definizione dei requisiti è importante avere una buona capacità di comprensione del linguaggio.

## Capacità di apprendimento successivo

Le basi dell'analisi di un problema per comprendere le specifiche e progettare sia le strutture dati necessarie che l'algoritmo più adatto è applicabile ad altri linguaggi di programmazione e potrà aiutare nei successivi corsi di programmazione.

1020420 | METODI  
MATEMATICI PER  
L'INFORMATICA

1°

6

ITA

**Insegnamento****Semestre****CFU****Lingua****Obiettivi formativi**

## Obiettivi generali:

L'insegnamento è indirizzato all'acquisizione delle conoscenze logiche e insiemistiche di base per affrontare lo studio di altri argomenti in matematica ed informatica.

## Obiettivi specifici:

## Conoscenza e comprensione:

Alla fine del corso lo studente avrà piena comprensione degli strumenti logico-insiemistici proposti

## Applicare conoscenza e comprensione:

Avrà acquisito la capacità di portare avanti un rigoroso, anche se elementare, ragionamento matematico, in particolare per quanto riguarda i principi logici fondamentali e l'uso dell'induzione in tutte le sue forme.

## Capacità critiche e di giudizio:

Sarà quindi capace di affrontare criticamente gli argomenti proposti in altri insegnamenti sia teorici che applicati. A questo proposito vengono forniti molti esempi presi da altri insegnamenti.

## Capacità di comunicare quanto si è appreso:

La partecipazione attiva in classe e l'uso dell'esame orale servono a stimolare lo studente nell'acquisire il linguaggio proprio della matematica e a trasmettere le conoscenze e le capacità acquisite in maniera appropriata.

Capacità di proseguire lo studio: Lo studente sarà in grado di approfondire nello studio personale i temi trattati usando quanto appreso come base.

1015880 |  
PROGETTAZIONE DI  
SISTEMI DIGITALI

1°

6

ITA

**Obiettivi formativi**

## Obiettivi generali:

Metodologie di progettazione di circuiti combinatori e sequenziali.

## Obiettivi specifici:

codifica binaria di vari tipi di dato, algebra booleana, analisi e sintesi di circuiti combinatori, flip-flop, analisi e sintesi di circuiti sequenziali, registri, interconnessione tra registri e altri moduli

## Conoscenza e comprensione:

Conoscere e capire come l'elaboratore gestisce ed elabora l'informazione

## Applicare conoscenza e comprensione:

Dimostrare di saper progettare semplici circuiti combinatori e sequenziali in grado di svolgere determinati compiti.

## Capacità critiche e di giudizio:

saper scegliere il miglior approccio, tra i vari studiati, per risolvere un determinato compito

## Capacità comunicative:

essere in grado di valutare e motivare le proprie scelte nella progettazione di un circuito

## Capacità di apprendimento:

Capire le differenze e i vantaggi delle varie tecniche di progettazione.

1015881 |  
ARCHITETTURA DEGLI  
ELABORATORI

2°

6

ITA

**Insegnamento****Semestre****CFU****Lingua****Obiettivi formativi**

## Obiettivi generali:

L'obiettivo dell'insegnamento di Architettura degli elaboratori è di far comprendere i principi che sono usati per progettare i calcolatori moderni. In particolare, il corso tratta la struttura interna del microprocessore e le idee che hanno permesso la straordinaria evoluzione della potenza di calcolo negli ultimi 30 anni. come pipelining, caching, branch prediction, e multi-processing.

## Obiettivi specifici:

Il corso tratta i principi di base di organizzazione del microprocessore e le nozioni di pipelining, caching, branch prediction, virtualizzazione e multi-processing. Inoltre, il corso tratta la programmazione assembly.

## Conoscenza e comprensione:

Lo studente acquisirà conoscenza sull'organizzazione del microprocessore MIPS, come implementazione delle idee generali che fanno parte degli obiettivi del corso. Inoltre, lo studente acquisirà conoscenza su come si strutturano i programmi in assembly, incluse le strutture dati, i paradigmi standard di programmazione e la ricorsione.

## Applicazione di conoscenza e comprensione:

Le conoscenze sono applicate sulla architettura MIPS, comprese in modo tale da poter capire le implicazioni delle scelte di programmazione sulla performance dei programmi su hardware specifico. Questo risultato è ottenuto tramite esercizi di programmazione e di valutazione delle prestazioni.

## Autonomia di giudizio:

Lo studente sarà in grado di comprendere le problematiche relative alle prestazioni del software su hardware specifico e di valutarne autonomamente le caratteristiche.

## Abilità comunicative:

Il corso non si propone espliciti obiettivi sulle capacità comunicative, eccetto di formare all'esposizione rigorosa degli argomenti tecnici.

## Capacità di apprendimento successivo:

Il corso pone le basi per la comprensione dei moduli dell'insegnamento di Sistemi operativi e di tutti i corsi di programmazione, inclusa la programmazione di sistemi paralleli.

97796 | CALCOLO  
INTEGRALE

2°

6

ITA

**Obiettivi formativi**

## Obiettivi generali:

Lo scopo del corso è fornire strumenti di analisi matematica di base come integrali, serie numeriche e di potenze ed equazioni differenziali.

## Obiettivi specifici:

## Conoscenza e comprensione:

L'obiettivo è che lo studente apprenda alcune tecniche di integrazione per calcolare integrali definiti ed indefiniti usando la formula di integrazione per parti e l'integrazione per sostituzione.

Lo studente sarà in grado di studiare la convergenza delle serie numeriche e approssimare, in alcuni casi, la loro somma e studierà le serie di potenze e gli sviluppi in serie di Taylor delle funzioni elementari. Infine imparerà a risolvere alcune equazioni differenziali ordinarie del primo ordine a variabili separabili e lineari del primo ordine ed equazioni differenziali del secondo ordine lineari con coefficienti costanti omogenee e non.

## Applicazione di conoscenza e comprensione:

Il corso prevede lezioni teoriche ed esercitazioni per apprendere lo svolgimento degli esercizi pratici.

A meta' del corso è prevista una prova pratica per verificare l'apprendimento del programma svolto.

1015885 |  
INTRODUZIONE AGLI  
ALGORITMI

2°

6

ITA

**Obiettivi formativi**

## Obiettivi generali

Introduzione ai metodi di base per la progettazione e l'analisi degli algoritmi, iterativi e ricorsivi, ed alla valutazione della loro efficienza computazionale.

## Obiettivi specifici

## Conoscenza e comprensione:

Al termine del corso gli studenti conosceranno le metodologie di base per la progettazione e l'analisi di algoritmi iterativi e ricorsivi, le strutture dati elementari, alcuni modi per scandire tali strutture, i principali algoritmi di ordinamento e le implementazioni più elementari dei dizionari.

## Applicare conoscenza e comprensione:

Al termine del corso gli studenti avranno acquisito familiarità con le principali strutture dati di base, in particolare quelle che implementano i dizionari. Sapranno spiegarne gli algoritmi e analizzarne la complessità, evidenziando come le prestazioni dipendano dalla struttura dati utilizzata. Saranno in grado di progettare nuove strutture dati e i relativi algoritmi, rielaborando quelli esistenti; sapranno spiegare i principali algoritmi di ordinamento, illustrando le strategie di progetto sottostanti e le relative analisi di complessità; saranno in grado di confrontare i comportamenti asintotici dei tempi di esecuzione degli algoritmi studiati; saranno in grado di progettare soluzioni ricorsive di problemi e di analizzare asintoticamente gli algoritmi risultanti.

## Capacità critiche e di giudizio:

Lo studente avrà le basi per analizzare la qualità di un algoritmo e delle relative strutture dati, sia dal punto di vista della effettiva risoluzione del problema che da quello della efficienza computazionale con la quale il problema viene risolto.

## Capacità comunicative:

Lo studente acquisirà la capacità di esporre in modo chiaro ed organizzato le proprie conoscenze, capacità che verrà verificata sia mediante i quesiti presentati nelle prove scritte che durante la prova orale. Lo studente sarà in grado di esprimere un'idea algoritmica in modo rigoroso ad alto livello, in pseudocodice.

## Capacità di apprendimento successivo:

Le conoscenze acquisite permetteranno allo studente di affrontare lo studio, individuale o previsto nell'ambito di un corso di laurea magistrale, di tecniche algoritmiche e di strutture dati di base.

**Insegnamento****Semestre****CFU****Lingua****Obiettivi formativi**

## Obiettivi generali:

Apprendimento dei concetti della programmazione orientata agli oggetti mediante il linguaggio di programmazione Java.

## Obiettivi specifici:

I concetti fondamentali della programmazione orientata agli oggetti: classi e oggetti, incapsulamento, ereditarietà, polimorfismo, binding statico e dinamico, i design pattern. La programmazione funzionale. Gli strumenti e le metodologie di base della progettazione software tramite un linguaggio orientato agli oggetti. Il linguaggio Java.

## Conoscenza e comprensione:

Conoscenza dei costrutti dei linguaggi di programmazione orientata agli oggetti, con particolare riferimento al linguaggio Java. Comprensione di un programma Java. Capacità di scrittura di un programma Java di piccole e medie dimensioni.

## Applicare conoscenza e comprensione:

Essere in grado di applicare le metodologie di base per affrontare la progettazione di sistemi software di grandezza medio-piccola. Saper usare i principali strumenti di sviluppo per realizzare tali sistemi in Java.

## Capacità critiche e di giudizio:

Capacità di identificare istruzioni, costrutti o pattern errati o inefficienti così come corretti o efficienti in Java.

## Capacità comunicative:

Illustrazione del progetto sviluppato.

## Capacità di apprendimento successivo:

Capacità di apprendere e applicare nuove tecniche di programmazione a partire da quelle apprese durante il corso.

AAF1101 | LINGUA  
INGLESE

2°

3

ENG

**Obiettivi formativi**

L'insegnamento della lingua inglese punta a mettere in grado lo studente di leggere correttamente la letteratura scientifica in inglese, la documentazione tecnica, e di poter sostenere conversazioni in inglese su aspetti tecnici.

**2° anno****Insegnamento**

1015886 | ALGEBRA

**Semestre**

1°

**CFU**

9

**Lingua**

ITA

**Obiettivi formativi**

## Obiettivi generali:

Fornire conoscenze di base di teoria dei gruppi e di algebra lineare che sono comunemente utilizzati in informatica."

## Obiettivi specifici:

Concetti di Algebra elementari

Introduzione alle strutture algebriche

Sviluppo del linguaggio dell'algebra lineare: spazi vettoriali e loro omomorfismi; spazi vettoriali, numeri e algebra delle matrici; endomorfismi, determinanti e diagonalizzazione; applicazioni.

## Conoscenze e comprensione:

Alla fine del corso lo studente sarà in grado di fare conti in gruppi piccoli di facile struttura e di comprendere il significato e l'utilizzo di matrici nello studio di fenomeni lineari.

## Applicare conoscenza e comprensione:

Utilizzo della diagonalizzazione di operatori per fornire un'interpretazione geometrica ad un fenomeno lineare e ricavarne informazioni rilevanti. Utilizzo del concetto di gruppo nello studio di problemi combinatori finiti.

## Capacità di giudizio:

Gli studenti saranno in grado di scegliere lo strumento algebrico più appropriato allo studio di un problema.

## Capacità di comunicazione:

Gli studenti impareranno il linguaggio dell'algebra lineare e dei gruppi e rudimenti di complessità.

## Capacità di apprendimento:

L'apprendimento del linguaggio lineare e grupppale fornirà agli studenti la capacità di apprendere argomenti che ne fanno uso.

1020421 | CALCOLO  
DELLE PROBABILITA'

1°

9

ITA

**Obiettivi formativi**

## Obiettivi generali:

Acquisire conoscenza e capacità di applicazione di argomenti di base di probabilità e statistica.

## Obiettivi specifici:

Assiomi e proprietà elementari delle probabilità. Variabili Aleatorie. Distribuzioni continue e discrete. Valori attesi.

Introduzione all'inferenza statistica.

## Conoscenza e comprensione:

Al termine del corso lo studente avrà acquisito le nozioni e i risultati di base relativi alla teoria della probabilità su spazi finiti e numerabili, al concetto di vettore aleatorio discreto e al concetto di variabile aleatoria continua.

## Applicare conoscenza e comprensione:

Gli studenti che abbiano superato l'esame saranno in grado di applicare le nozioni di calcolo combinatorio per risolvere semplici problemi di probabilità discreta, problemi inerenti vettori casuali discreti e numeri casuali rappresentati da variabili aleatorie continue. Lo studente sarà anche in grado di apprezzare il significato e le implicazioni dell'indipendenza e del condizionamento (nell'ambito di modelli discreti), comprendere il significato di alcuni teoremi limite fondamentali, quali la legge dei grandi numeri.

## Capacità critiche e di giudizio:

Lo studente avrà le basi per analizzare e costruire modelli probabilistici in semplici situazioni di interesse fisico, biologico e tecnologico, utilizzare tavole e software di simulazione delle leggi discrete di più comune applicazione, nonché della legge gaussiana, e di comprendere l'utilizzazione di strumenti statistici elementari nell'inferenza, nel campionamento statistico e nella simulazione.

## Capacità comunicative:

Capacità di esporre i contenuti nella parte orale della verifica e negli eventuali quesiti teorici presenti nella prova scritta.

Capacità di apprendimento: le conoscenze acquisite permetteranno uno studio, individuale o impartito in un corso relativo ad aspetti più specialistici di teoria della probabilità.

Insegnamento	Semestre	CFU	Lingua
1020422   SISTEMI OPERATIVI	1°	12	ITA

### Obiettivi formativi

#### Obiettivi generali:

Il corso ha come obiettivi i concetti, la struttura, e i meccanismi dei sistemi operativi. Verranno trattate caratteristiche fondamentali, presenti fin dai sistemi più tradizionali, ma anche peculiarità dei sistemi moderni che nascono come conseguenza dell'evoluzione ricorrente della tecnologia.

Conoscere ed usare l'interfaccia di programmazione fra sviluppatore software e kernel relativamente ai servizi base di accesso alle risorse del sistema operativo Linux.

#### Obiettivi specifici:

Caratteristiche e concetti dei sistemi operativi moderni, con particolare riferimento ai sistemi Unix e Linux. Si inizierà con una descrizione dell'evoluzione dei sistemi operativi nel tempo, per continuare con concetti fondamentali come i processi, lo stallo, e relativi meccanismi di prevenzione, la concorrenza tra processi, la gestione della memoria, processore e I/O, i file system e la sicurezza.

Conoscere i concetti (processi, espressioni regolari, file system) e i comandi shell più importanti di Linux. Saper creare script in Bash. Saper scrivere programmi C che usino le system call di Linux.

#### Conoscenza e comprensione:

Capire in modo profondo come i sistemi operativi danno supporto all'esecuzione dei programmi degli utenti e gestiscono le periferiche hardware di un computer. Metodi e tecniche fondamentali per la rappresentazione dei processi in memoria e la gestione efficiente di multiprogrammazione—molteplici processi eseguiti contemporaneamente in un sistema con risorse limitate.

Conoscenza del funzionamento interno del sistema operativo Linux. Conoscenza del funzionamento della shell Bash. Fondamenti del linguaggio C. Conoscenza delle principali system call di Linux.

#### Applicare conoscenza e comprensione:

Progettare programmi a livello utente e di sistema in modo efficiente e sicuro.

Saper creare script in Bash in grado di risolvere problemi pratici. Saper scrivere programmi in C che sfruttino le system call di Linux per ottimizzare l'uso delle risorse.

#### Capacità critiche e di giudizio:

Essere in grado di predire l'uso delle risorse richieste da un programma, di scoprire una possibile situazione di stallo in un sistema multiprogrammato, garantire la mutua esclusione tra processi e l'accesso protetto a zone di memoria o risorse sensibili.

Saper valutare la soluzione più appropriata per ottenere un determinato risultato, usando o singoli comandi shell, o uno script Bash, o un programma C basato su system call di Linux.

#### Capacità comunicative:

Saper comunicare in modo chiaro e preciso le caratteristiche dei sistemi operativi e i loro meccanismi di supporto software/hardware.

Essere in grado di comunicare e documentare script Bash e programmi C basati su system call di Linux.

#### Capacità di apprendimento:

Saper sfruttare la conoscenza acquisita nella progettazione di sistemi e di programmi utente nel modulo successivo del corso. Essere in grado di usare questa conoscenza nell'apprendimento di proprietà di sistemi più complessi come quelli distribuiti e cloud.

Essere in grado di usare i concetti appresi in corsi avanzati da sistemista aziendale, o in un eventuale corso avanzato che richieda interazione con Linux, come ad es.: programmazione di sistema, cloud computing, sistemi distribuiti, cybersecurity.

I MODULO

1°

6

ITA

**Obiettivi formativi****Obiettivi generali:**

Il corso ha come obiettivi i concetti, la struttura, e i meccanismi dei sistemi operativi. Verranno trattate caratteristiche fondamentali, presenti fin dai sistemi più tradizionali, ma anche peculiarità dei sistemi moderni che nascono come conseguenza dell'evoluzione ricorrente della tecnologia.

**Obiettivi specifici:**

Il corso coprirà le caratteristiche e concetti dei sistemi operativi moderni, con particolare riferimento ai sistemi Unix e Linux. Si inizierà con una descrizione dell'evoluzione dei sistemi operativi nel tempo, per continuare con concetti fondamentali come i processi, lo stallo, e relativi meccanismi di prevenzione, la concorrenza tra processi, la gestione della memoria, processore e I/O, i file system e la sicurezza.

**Conoscenza e comprensione:**

Capire in modo profondo come i sistemi operativi danno supporto all'esecuzione dei programmi degli utenti e gestiscono le periferiche hardware di un computer. Metodi e tecniche fondamentali per la rappresentazione dei processi in memoria e la gestione efficiente di multiprogrammazione—molteplici processi eseguiti contemporaneamente in un sistema con risorse limitate.

**Applicare conoscenza e comprensione:**

Progettare programmi a livello utente e di sistema in modo efficiente e sicuro.

**Capacità critiche e di giudizio:**

Essere in grado di predire l'uso delle risorse richieste da un programma, di scoprire una possibile situazione di stallo in un sistema multiprogrammato, garantire la mutua esclusione tra processi e l'accesso protetto a zone di memoria o risorse sensibili.

**Capacità comunicative:**

Saper comunicare in modo chiaro e preciso le caratteristiche dei sistemi operativi e i loro meccanismi di supporto software/hardware.

**Capacità di apprendimento:**

Saper sfruttare la conoscenza acquisita nella progettazione di sistemi e di programmi utente nel modulo successivo del corso. Essere in grado di usare questa conoscenza nell'apprendimento di proprietà di sistemi più complessi come quelli distribuiti e cloud.

**Insegnamento****Semestre****CFU****Lingua****Obiettivi formativi**

## Obiettivi generali:

Esporre gli studenti a solide metodologie di progettazione e realizzazione di applicazioni per basi di dati

## Obiettivi specifici:

Esporre gli studenti a metodologie formali e scalabili per l'analisi concettuale e la progettazione di applicazioni per basi di dati e a tecnologie standard per la loro realizzazione.

## Conoscenza e comprensione:

Gli studenti acquisiranno conoscenze metodologiche fondamentali per la progettazione di applicazioni per basi di dati non banali (in particolare per le fasi di: a) raccolta dei requisiti; b) analisi concettuale dei dati e delle funzionalità; c) progettazione della base dati e delle funzionalità) e per la loro realizzazione (utilizzo di DBMS e di linguaggi standard di definizione, interrogazione e manipolazione dei dati).

## Applicare conoscenza e comprensione:

Gli studenti saranno in grado di applicare in modo efficace le conoscenze indicate al punto precedente in progetti reali di applicazioni per basi di dati non banali.

## Capacità critiche e di giudizio:

Gli studenti saranno in grado di prendere autonomamente decisioni razionali in tutte le fasi del processo di progettazione di applicazioni per basi di dati.

## Capacità comunicative:

Gli studenti saranno in grado di interagire in modo proficuo con i committenti (per quanto concerne la raccolta dei requisiti) e con altri analisti e progettisti (per quanto concerne le attività di analisi e progettazione di sistemi software non banali).

## Capacità di apprendimento:

Gli studenti saranno in grado di ampliare le loro conoscenze in modo autonomo consultando, secondo necessità, manualistica tecnica nell'ambito della progettazione di applicazioni per basi di dati.

I MODULO

1°

6

ITA

**Obiettivi formativi**

## Obiettivi generali:

Esporre gli studenti a solide metodologie di progettazione e realizzazione di applicazioni per basi di dati

## Obiettivi specifici:

Esporre gli studenti a metodologie formali e scalabili per l'analisi concettuale e la progettazione di applicazioni per basi di dati e a tecnologie standard per la loro realizzazione.

## Conoscenza e comprensione:

Gli studenti acquisiranno conoscenze metodologiche fondamentali per la progettazione di applicazioni per basi di dati non banali (in particolare per le fasi di: a) raccolta dei requisiti; b) analisi concettuale dei dati e delle funzionalità; c) progettazione della base dati e delle funzionalità) e per la loro realizzazione (utilizzo di DBMS e di linguaggi standard di definizione, interrogazione e manipolazione dei dati).

## Applicare conoscenza e comprensione:

Gli studenti saranno in grado di applicare in modo efficace le conoscenze indicate al punto precedente in progetti reali di applicazioni per basi di dati non banali.

## Capacità critiche e di giudizio:

Gli studenti saranno in grado di prendere autonomamente decisioni razionali in tutte le fasi del processo di progettazione di applicazioni per basi di dati.

## Capacità comunicative:

Gli studenti saranno in grado di interagire in modo proficuo con i committenti (per quanto concerne la raccolta dei requisiti) e con altri analisti e progettisti (per quanto concerne le attività di analisi e progettazione di sistemi software non banali).

## Capacità di apprendimento:

Gli studenti saranno in grado di ampliare le loro conoscenze in modo autonomo consultando, secondo necessità, manualistica tecnica nell'ambito della progettazione di applicazioni per basi di dati.

Insegnamento	Semestre	CFU	Lingua
1015887   BASI DI DATI	2°	12	ITA

### Obiettivi formativi

#### Obiettivi generali:

Esporre gli studenti a solide metodologie di progettazione e realizzazione di applicazioni per basi di dati

#### Obiettivi specifici:

Esporre gli studenti a metodologie formali e scalabili per l'analisi concettuale e la progettazione di applicazioni per basi di dati e a tecnologie standard per la loro realizzazione.

#### Conoscenza e comprensione:

Gli studenti acquisiranno conoscenze metodologiche fondamentali per la progettazione di applicazioni per basi di dati non banali (in particolare per le fasi di: a) raccolta dei requisiti; b) analisi concettuale dei dati e delle funzionalità; c) progettazione della base dati e delle funzionalità) e per la loro realizzazione (utilizzo di DBMS e di linguaggi standard di definizione, interrogazione e manipolazione dei dati).

#### Applicare conoscenza e comprensione:

Gli studenti saranno in grado di applicare in modo efficace le conoscenze indicate al punto precedente in progetti reali di applicazioni per basi di dati non banali.

#### Capacità critiche e di giudizio:

Gli studenti saranno in grado di prendere autonomamente decisioni razionali in tutte le fasi del processo di progettazione di applicazioni per basi di dati.

#### Capacità comunicative:

Gli studenti saranno in grado di interagire in modo proficuo con i committenti (per quanto concerne la raccolta dei requisiti) e con altri analisti e progettisti (per quanto concerne le attività di analisi e progettazione di sistemi software non banali).

#### Capacità di apprendimento:

Gli studenti saranno in grado di ampliare le loro conoscenze in modo autonomo consultando, secondo necessità, manualistica tecnica nell'ambito della progettazione di applicazioni per basi di dati.

II MODULO

2°

6

ITA

**Obiettivi formativi**

## Obiettivi generali:

Essere in grado di progettare/valutare le proprietà, la struttura di memorizzazione e i protocolli di un sistema di gestione di basi di dati.

Esporre gli studenti a solide metodologie di progettazione e realizzazione di applicazioni per basi di dati

## Obiettivi specifici:

Conoscenza delle proprietà di uno schema relazionale e di una decomposizione. Capacità di interrogare una base di dati.

Capacità di valutare i costi delle operazioni di accesso ai dati. Conoscenza di protocolli di per il controllo della concorrenza.

Esporre gli studenti a metodologie formali e scalabili per l'analisi concettuale e la progettazione di applicazioni per basi di dati e a tecnologie standard per la realizzazione di basi di dati relazionali.

## Conoscenza e comprensione:

Fondamenti teorici della progettazione e della interrogazione di una base di dati relazionale. Principali strutture di organizzazione dei dati su memoria secondaria. Principali tecniche usate nei DBMS per il controllo dell'esecuzione concorrente delle transazioni.

Gli studenti acquisiranno conoscenze metodologiche fondamentali per la progettazione di applicazioni per basi di dati non banali (in particolare per le fasi di: a) raccolta dei requisiti; b) analisi concettuale dei dati e delle funzionalità; c) progettazione della base dati e delle funzionalità) e per la loro realizzazione (utilizzo di DBMS relazionali e del linguaggio SQL).

## Applicare conoscenza e comprensione:

Progettare schemi relazionali con "buone proprietà". Interrogare una base di dati mediante algebra relazionale. Valutare i costi delle operazioni fondamentali su file con diversi tipi di organizzazione fisica.

Gli studenti saranno in grado di applicare in modo efficace le conoscenze acquisite in progetti reali di applicazioni per basi non banali.

## Capacità critiche e di giudizio:

Essere in grado di valutare le proprietà di uno schema relazionale e di una sua decomposizione. Essere in grado di scegliere la struttura dati più appropriata per memorizzare le informazioni di uno schema.

Gli studenti saranno in grado di prendere autonomamente decisioni razionali in tutte le fasi del processo di progettazione di applicazioni per basi di dati.

## Capacità comunicative:

Essere in grado di comunicare/condividere caratteristiche qualitative/quantitative relative alla struttura relazionale di una base di dati

Gli studenti saranno in grado di interagire in modo proficuo con i committenti (per quanto concerne la raccolta dei requisiti) e con altri analisti e progettisti (per quanto concerne le attività di analisi e progettazione di sistemi software non banali).

## Capacità di apprendimento:

Essere in grado di utilizzare i concetti acquisiti nel modulo successivo del corso e in un eventuale corso avanzato di basi di Dati.

Gli studenti saranno in grado di ampliare le loro conoscenze in modo autonomo consultando, secondo necessità, manualistica tecnica nell'ambito della progettazione di applicazioni per basi di dati.

**Insegnamento****Semestre****CFU****Lingua****Obiettivi formativi**

## Obiettivi generali

Acquisire la conoscenza di base delle più note tecniche algoritmiche di progettazione e delle tecniche di valutazione della correttezza e della complessità degli algoritmi.

## Obiettivi specifici

## Conoscenza e comprensione:

Al termine del corso gli studenti posseggono le conoscenze di base relative a:

- tecniche fondamentali di progettazione algoritmica;
- analisi della correttezza e della efficienza degli algoritmi;

## Applicazione di conoscenza e comprensione:

Al termine del corso gli studenti sono in grado di:

- analizzare le prestazioni di un algoritmo tramite strumenti matematici rigorosi;
- analizzare algoritmi e strutture dati
- progettare ed analizzare nuovi algoritmi, sfruttando le metodologie presentate durante il corso.

## Autonomia di giudizio:

Lo studente alla fine del corso deve essere in grado di scegliere autonomamente qual'è la tecnica algoritmica più adatta da applicare per un determinato problema e valutare tra più soluzioni algoritmiche per un certo problema qual'è da preferirsi.

## Abilità comunicative:

Lo studente acquisirà la capacità di esprimere un'idea algoritmica tramite l'uso di uno pseudocodice.

## Capacità di apprendimento:

Lo studente avrà acquisito la capacità di analizzare un problema, progettare le necessarie strutture dati e un algoritmo corretto ed efficiente che lo risolva.

1015889 | RETI DI  
ELABORATORI

2°

9

ITA

**Obiettivi formativi**

## Obiettivi generali:

Imparare le basi delle reti di elaboratori moderne e Internet.

## Obiettivi specifici:

## Conoscenza e comprensione:

Conoscenza e comprensione dei protocolli dei livelli applicazione, trasporto, rete e collegamento della pila TCP/IP.

## Applicazione di conoscenza e comprensione:

Capacità di comprendere qualsiasi protocollo coinvolto nella comunicazione TCP/IP

Capacità di comprendere strumenti e tecniche per risolvere problemi di rete

Capacità di usare servizi di rete come DNS e DHCP che permettono di far funzionare una rete

## Autonomia di giudizio:

Individuare problemi di rete

Valutare la realizzazione di nuovi servizi

## Abilità comunicative:

Capacità di descrivere le reti in termini strutturali secondo il modello a 5 livelli.

## Capacità di apprendimento successivo:

Il corso fornisce le basi per poter apprendere nozioni di reti wireless e Internet of Things.

1020422 | SISTEMI  
OPERATIVI

2°

12

ITA

**Obiettivi formativi**

## Obiettivi generali:

Il corso ha come obiettivi i concetti, la struttura, e i meccanismi dei sistemi operativi. Verranno trattate caratteristiche fondamentali, presenti fin dai sistemi più tradizionali, ma anche peculiarità dei sistemi moderni che nascono come conseguenza dell'evoluzione ricorrente della tecnologia.

Conoscere ed usare l'interfaccia di programmazione fra sviluppatore software e kernel relativamente ai servizi base di accesso alle risorse del sistema operativo Linux.

## Obiettivi specifici:

Caratteristiche e concetti dei sistemi operativi moderni, con particolare riferimento ai sistemi Unix e Linux. Si inizierà con una descrizione dell'evoluzione dei sistemi operativi nel tempo, per continuare con concetti fondamentali come i processi, lo stallo, e relativi meccanismi di prevenzione, la concorrenza tra processi, la gestione della memoria, processore e I/O, i file system e la sicurezza.

Conoscere i concetti (processi, espressioni regolari, file system) e i comandi shell più importanti di Linux. Saper creare script in Bash. Saper scrivere programmi C che usino le system call di Linux.

## Conoscenza e comprensione:

Capire in modo profondo come i sistemi operativi danno supporto all'esecuzione dei programmi degli utenti e gestiscono le periferiche hardware di un computer. Metodi e tecniche fondamentali per la rappresentazione dei processi in memoria e la gestione efficiente di multiprogrammazione—molteplici processi eseguiti contemporaneamente in un sistema con risorse limitate.

Conoscenza del funzionamento interno del sistema operativo Linux. Conoscenza del funzionamento della shell Bash. Fondamenti del linguaggio C. Conoscenza delle principali system call di Linux.

## Applicare conoscenza e comprensione:

Progettare programmi a livello utente e di sistema in modo efficiente e sicuro.

Saper creare script in Bash in grado di risolvere problemi pratici. Saper scrivere programmi in C che sfruttino le system call di Linux per ottimizzare l'uso delle risorse.

## Capacità critiche e di giudizio:

Essere in grado di predire l'uso delle risorse richieste da un programma, di scoprire una possibile situazione di stallo in un sistema multiprogrammato, garantire la mutua esclusione tra processi e l'accesso protetto a zone di memoria o risorse sensibili.

Saper valutare la soluzione più appropriata per ottenere un determinato risultato, usando o singoli comandi shell, o uno script Bash, o un programma C basato su system call di Linux.

## Capacità comunicative:

Saper comunicare in modo chiaro e preciso le caratteristiche dei sistemi operativi e i loro meccanismi di supporto software/hardware.

Essere in grado di comunicare e documentare script Bash e programmi C basati su system call di Linux.

## Capacità di apprendimento:

Saper sfruttare la conoscenza acquisita nella progettazione di sistemi e di programmi utente nel modulo successivo del corso. Essere in grado di usare questa conoscenza nell'apprendimento di proprietà di sistemi più complessi come quelli distribuiti e cloud.

Essere in grado di usare i concetti appresi in corsi avanzati da sistemista aziendale, o in un eventuale corso avanzato che richieda interazione con Linux, come ad es.: programmazione di sistema, cloud computing, sistemi distribuiti, cybersecurity.

**Insegnamento****Semestre****CFU****Lingua****Obiettivi formativi**

## Obiettivi generali:

Il corso ha come obiettivi i concetti, la struttura, e i meccanismi dei sistemi operativi. Verranno trattate caratteristiche fondamentali, presenti fin dai sistemi più tradizionali, ma anche peculiarità dei sistemi moderni che nascono come conseguenza dell'evoluzione ricorrente della tecnologia.

Conoscere ed usare l'interfaccia di programmazione fra sviluppatore software e kernel relativamente ai servizi base di accesso alle risorse del sistema operativo Linux.

## Obiettivi specifici:

Caratteristiche e concetti dei sistemi operativi moderni, con particolare riferimento ai sistemi Unix e Linux. Si inizierà con una descrizione dell'evoluzione dei sistemi operativi nel tempo, per continuare con concetti fondamentali come i processi, lo stallo, e relativi meccanismi di prevenzione, la concorrenza tra processi, la gestione della memoria, processore e I/O, i file system e la sicurezza.

Conoscere i concetti (processi, espressioni regolari, file system) e i comandi shell più importanti di Linux. Saper creare script in Bash. Saper scrivere programmi C che usino le system call di Linux.

## Conoscenza e comprensione:

Capire in modo profondo come i sistemi operativi danno supporto all'esecuzione dei programmi degli utenti e gestiscono le periferiche hardware di un computer. Metodi e tecniche fondamentali per la rappresentazione dei processi in memoria e la gestione efficiente di multiprogrammazione—molteplici processi eseguiti contemporaneamente in un sistema con risorse limitate.

Conoscenza del funzionamento interno del sistema operativo Linux. Conoscenza del funzionamento della shell Bash.

Fondamenti del linguaggio C. Conoscenza delle principali system call di Linux.

## Applicare conoscenza e comprensione:

Progettare programmi a livello utente e di sistema in modo efficiente e sicuro.

Saper creare script in Bash in grado di risolvere problemi pratici. Saper scrivere programmi in C che sfruttino le system call di Linux per ottimizzare l'uso delle risorse.

## Capacità critiche e di giudizio:

Essere in grado di predire l'uso delle risorse richieste da un programma, di scoprire una possibile situazione di stallo in un sistema multiprogrammato, garantire la mutua esclusione tra processi e l'accesso protetto a zone di memoria o risorse sensibili.

Saper valutare la soluzione più appropriata per ottenere un determinato risultato, usando o singoli comandi shell, o uno script Bash, o un programma C basato su system call di Linux.

## Capacità comunicative:

Saper comunicare in modo chiaro e preciso le caratteristiche dei sistemi operativi e i loro meccanismi di supporto software/hardware.

Essere in grado di comunicare e documentare script Bash e programmi C basati su system call di Linux.

## Capacità di apprendimento:

Saper sfruttare la conoscenza acquisita nella progettazione di sistemi e di programmi utente nel modulo successivo del corso. Essere in grado di usare questa conoscenza nell'apprendimento di proprietà di sistemi più complessi come quelli distribuiti e cloud.

Essere in grado di usare i concetti appresi in corsi avanzati da sistemista aziendale, o in un eventuale corso avanzato che richieda interazione con Linux, come ad es.: programmazione di sistema, cloud computing, sistemi distribuiti, cybersecurity.

**3° anno****Insegnamento****Semestre****CFU****Lingua**

1022301 | INGEGNERIA  
DEL SOFTWARE

1°

6

ITA

**Insegnamento****Semestre****CFU****Lingua****Obiettivi formativi****Obiettivi generali:**

Il corso illustra i fondamenti delle metodologie e degli strumenti per la gestione dei processi software. Particolare attenzione viene dedicata alle metodologie di analisi e progettazione orientate agli oggetti, e alla loro gestione e documentazione mediante UML.

**Obiettivi specifici:**

Introduzione agli approcci all'ingegneria del software e al ciclo di vita del software, approfondimento sulle attività di specifica, analisi, progetto e test di sistemi software, tecniche per la gestione dei processi, con particolare riferimento alla gestione della qualità e dei rischi e all'analisi dei costi.

**Conoscenze e comprensione:**

Alla fine del corso lo studente avrà acquisito conoscenze relative ai principali modelli di ciclo di vita del software, alle metriche per il dimensionamento dello sforzo, alle tecniche di descrizione delle diverse componenti di un progetto software. Saranno state acquisite conoscenze relative all'uso del linguaggio UML. Infine, avranno anche acquisito familiarità con la letteratura scientifica nel campo.

**Applicarei conoscenza e comprensione:**

Alla fine del corso lo studente sarà in grado di lavorare in team alle attività di analisi, progettazione, documentazione e gestione di progetti software di medie dimensioni. Avranno imparato a produrre documentazione basata su UML, relativamente ai principali tipi di diagrammi: dei casi d'uso, delle classi, di interazione, di stato e di attività, anche attraverso l'utilizzo di ambienti software professionali orientati allo sviluppo sistematico di progetti software. Infine saranno in grado di produrre una valutazione dello sforzo basata su Punti Funzione e Punti Use Case.

**Capacità di giudizio:**

Gli studenti svilupperanno le capacità di analisi necessarie per valutare diverse alternative durante il processo di sviluppo software, con particolare riferimento alla valutazione delle scelte architetture e dei rischi di progetto.

**Capacità di comunicazione:**

Gli studenti impareranno a documentare le loro scelte, anche attraverso l'uso di strumenti di generazione della documentazione, in particolare sfruttando notazioni diagrammatiche. Avranno anche acquisito la capacità di preparare presentazioni relative ad argomenti scientifici.

**Capacità di apprendimento successivo:**

La conoscenza degli aspetti di rigore formale alla base della disciplina dell'ingegneria del software permetterà agli studenti di acquisire rapidamente confidenza con tecniche, oltre a quelle considerate nel corso, basate su principi generali.

1041727 | AUTOMI  
CALCOLABILITA' E  
COMPLESSITA'

1°

6

ITA

**Insegnamento****Semestre****CFU****Lingua****Obiettivi formativi**

## Obiettivi generali:

Durante il corso saranno introdotti i più importanti risultati dell'Informatica teorica: a partire dai fondamentali risultati in teoria della calcolabilità degli anni trenta, passando per quelli in teoria degli automi degli anni cinquanta per arrivare al problema aperto P contenuto o uguale a NP, esplicitamente sollevato negli anni settanta.

## Obiettivi specifici:

Gli studenti capiranno che ci sono diversi modelli di computazione e cosa ne determina il potere computazionale.

Gli studenti apprenderanno concetti astratti come classi di linguaggi, macchine universali, riducibilità e sapranno che alcuni problemi non possono essere risolti con un calcolatore e che altri sono computazionalmente difficili da risolvere o addirittura così difficili da poter essere considerati non risolvibili. Faremo vedere come alcuni di questi risultati sono utilizzati oggi.

## Conoscenza e comprensione:

Al termine del corso gli studenti conosceranno i metodi e risultati di base della della teoria degli automi, della calcolabilità e della complessità e sapranno applicarli per individuare la complessità di problemi in diversi campi. In particolare sapranno:

dimostrare l'equivalenza tra le diverse caratterizzazioni dei linguaggi regolari

dimostrare l'equivalenza tra le diverse caratterizzazioni dei linguaggi context-free

spiegare il concetto di non determinismo

giustificare l'esistenza di problemi privi di soluzioni algoritmiche o intrattabili.

## Applicazione di conoscenza e comprensione:

Gli studenti impareranno:

come costruire automi finiti (deterministici e non) da una specifica (formale o informale)

come costruire automi a pila (deterministici e non) da una specifica (formale o informale)

a usare la riducibilità tra problemi per dimostrarne la decidibilità o l'indecidibilità

a usare la riducibilità polinomiale per provare la NP-hardness di un problema

## Autonomia di giudizio:

Capire il giusto livello di astrazione utile per risolvere un problema, scegliere il modello computazionale più conveniente in un determinato contesto applicativo

## Abilità comunicative:

descrivere un linguaggio formale, a parole o attraverso uno degli strumenti offerti di descrizione finita, descrivere problemi indecidibili, intrattabili o trattabili, spiegare il significato e la rilevanza delle classi P ed NP nonché del problema "P=NP?"

## Capacità di apprendimento:

Lo studente sarà in grado di imparare altri modelli computazionali, sia completamente diversi da quelli studiati durante il corso, sia variazioni di questi. Egli sarà capace di capire nuove prove di NP-completezza o più in generale prove di completezza per una qualunque classe di complessità

A SCELTA DELLO  
STUDENTE

1°

6

ITA

A SCELTA DELLO  
STUDENTE

2°

6

ITA

AAF1004 | PROVA  
FINALE

2°

6

ITA

**Obiettivi formativi**

La prova finale consiste nella stesura, nella presentazione e nella discussione di una dissertazione scritta, elaborata autonomamente dallo studente, che documenti in modo organico e dettagliato il problema affrontato nell'ambito del tirocinio formativo e tutte le attività compiute per pervenire alla soluzione.

AAF1053 | TIROCINIO

2°

15

ITA

**Insegnamento****Semestre****CFU****Lingua****Obiettivi formativi**

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

Insegnamenti  
metodologici di  
completamento

Insegnamenti tecnologici  
affini

Insegnamenti tecnologici  
di completamento

**Gruppi opzionali**

Lo studente deve acquisire 6 CFU fra i seguenti esami

<b>Insegnamento</b>	<b>Anno</b>	<b>Semestre</b>	<b>CFU</b>	<b>Lingua</b>
1022264   LINGUAGGI DI PROGRAMMAZION E	3°	1°	6	ITA

**Obiettivi formativi**

Obiettivi generali:

Il corso illustra i fondamenti dei linguaggi di programmazione. Particolare attenzione viene dedicata alla semantica statica e dinamica dei linguaggi (teoria dei tipi, semantica operativa ed assiomatica) ed agli strumenti formali per la specifica e la verifica del software.

Obiettivi specifici:

Approfondimento sulle algebre induttive come fondamento di sintassi astratta, strutture dati e semplici logiche (equazionali) di programmi. Cenni sulla co-induzione come fondamento di analisi di processi concorrenti e strutture lazy. Confronto fra linguaggi lazy e eager, statici e dinamici, nel contesto di diversi paradigmi di programmazione: imperativo, funzionale, ad oggetti. Teoria dei tipi, con particolare attenzione al polimorfismo parametrico (alla ML) e di sottoclasse (alla Java).

Conoscenze e comprensione:

Alla fine del corso lo studente avrà acquisito conoscenze fondazionali relative ai linguaggi di programmazione e ai principali paradigmi di programmazione, e sarà in grado di inquadrare in una mappa concettuale coerente alcuni fra le principali aree di studio dell'informatica teorica, comprendendone le interconnessioni.

Applicazione di conoscenza e comprensione:

Alla fine del corso lo studente sarà in grado di orientarsi fra i diversi paradigmi della programmazione, con particolare attenzione alla programmazione funzionale. Avrà acquisito una conoscenza di base del linguaggio funzionale SML e della sua teoria dei tipi, nonché dell'uso dei metodi formali.

Capacità di giudizio:

Gli studenti acquisiranno strumenti culturali per sviluppare software di qualità e per comprendere ed inquadrare le tendenze emergenti dei linguaggi moderni.

Capacità di comunicazione:

Gli studenti impareranno ad usare il linguaggio della matematica e della logica per descrivere, sviluppare ed analizzare il software, e ad integrare l'approccio formale con un'esposizione intuitiva delle idee.

Capacità di apprendimento:

L'uso di strumenti matematici nella teoria e nella pratica della programmazione sviluppa capacità logiche e consapevolezza della complessità della materia di studio.

Insegnamento	Anno	Semestre	CFU	Lingua
10593235   PROGRAMMAZION E DI SISTEMI EMBEDDED E MULTICORE	3°	1°	6	ITA

#### Obiettivi formativi

##### Obiettivi generali:

Il corso è finalizzato all'acquisizione della conoscenza di tecniche algoritmiche e di programmazione adatte alle moderne piattaforme di calcolo multicore ed embedded.

##### Obiettivi specifici:

##### Conoscenza e comprensione:

Al termine del corso gli studenti avranno una profonda comprensione dei principali problemi di programmazione posti dai sistemi embedded e multi-core, nonché delle tecniche di programmazione utili per scrivere codice efficiente e scalabile. Avranno una comprensione della programmazione a basso livello e della progettazione HW.

##### Capacità di applicare conoscenza e comprensione:

gli studenti saranno in grado di progettare, programmare e realizzare sistemi embedded ed avranno una comprensione dei sistemi operativi adottati in tali tipi di sistemi.

##### Capacità critiche e di giudizio:

gli studenti saranno in grado di affrontare le sfide che si presentano nella progettazione di sistemi embedded e multicore e nell'implementazione di programmi efficienti e scalabili per tali sistemi, anche tenendo conto delle caratteristiche architetturali delle diverse piattaforme di calcolo.

##### Capacità di comunicare quanto si è appreso:

gli studenti saranno in grado di comunicare in modo efficace, riassumendo le idee principali nella progettazione di software embedded e multicore in modo chiaro e presentando informazioni tecniche accurate.

##### Capacità di studio autonomo:

obiettivo del corso è di introdurre i principi di base nella progettazione di software embedded e multicore. Gli studenti saranno quindi in grado di estendere la loro conoscenza in modo indipendente, adattandola all'evoluzione delle tecnologie e delle piattaforme di calcolo.

97597   FISICA	3°	1°	6	ITA
----------------	----	----	---	-----

#### Obiettivi formativi

##### Conoscenze acquisite

L'obiettivo principale del corso è di insegnare il metodo scientifico. Per ogni argomento trattato, sono presentate le osservazioni sperimentali (o numeriche) del fenomeno che si vuole capire. Viene quindi illustrato il processo di riduzionismo, con le relative approssimazioni, che porta ad un modello sufficientemente semplice da poter essere trattato matematicamente. Infine sono introdotte le leggi fisiche che permettono la spiegazione del fenomeno osservato, mettendo in luce l'ambito di validità di tale leggi e i loro limiti.

##### Competenze acquisite

Alla fine di questo corso lo studente dovrebbe aver appreso il metodo scientifico e dovrebbe essere in grado di applicarlo a qualsiasi problema di tipo scientifico che si trovi ad affrontare. In pratica dovrebbe aver sviluppato quella capacità di "problem solving" che è molto richiesta in qualsiasi disciplina tecnico-scientifica. Inoltre dovrebbe aver acquisito un bagaglio di leggi fisiche di base, che gli permettano di capire molti dei fenomeni naturali e dei dispositivi tecnologici che lo circondano.

10606972   DATA SCIENCE E DIRITTO	3°	1°	6	ITA
---	----	----	---	-----

1022262   INTELLIGENZA ARTIFICIALE	3°	2°	6	ITA
--	----	----	---	-----

Insegnamento	Anno	Semestre	CFU	Lingua
<b>Obiettivi formativi</b>				
Obiettivi generali: Il corso si propone di esporre gli studenti ad un'ampia panoramica dell'Intelligenza Artificiale (IA).				
Obiettivi specifici: Il corso si propone di mettere gli studenti in grado di comprendere la teoria di un'ampia gamma di tecniche di IA, e di padroneggiarle nella pratica della progettazione di sistemi software intelligenti.				
Conoscenza e comprensione: Introduzione ad ampio spettro dei principi fondamentali e delle diverse branche dell'Intelligenza Artificiale (IA) e conoscenze su risoluzione di problemi mediante ricerca, inferenza logica, pianificazione, ragionamento, apprendimento.				
Applicare conoscenza e comprensione: Saper applicare il portafoglio di tecniche e gli approcci illustrati per la progettazione e realizzazione di sistemi software intelligenti.				
Capacità critiche e di giudizio: Gli studenti saranno in grado di prendere autonomamente decisioni razionali sulle tecniche di IA migliori da impiegare nella progettazione di sistemi software intelligenti.				
Capacità comunicative: Gli studenti saranno in grado di interagire in modo proficuo con altri ricercatori in IA su un'ampia gamma di argomenti.				
Capacità di apprendimento: Gli studenti saranno in grado di ampliare le loro conoscenze in modo autonomo consultando, secondo necessità, la letteratura scientifica sull'IA.				
1022265   MODELLI E OTTIMIZZAZIONE	3°	2°	6	ITA

Insegnamento	Anno	Semestre	CFU	Lingua
<b>Obiettivi formativi</b>				
Obiettivi generali: Il corso introduce lo studente alla modellazione matematica di problemi di ottimizzazione, allo studio della programmazione lineare e intera e alle loro applicazioni in contesti reali.				
Obiettivi specifici:				
Apprendere:				
<ol style="list-style-type: none"> <li>1. Le principali tecniche di modellazione di problemi usando strumenti logico/matematici;</li> <li>2. Le proprietà teoriche e sulle applicazioni pratiche dei principali problemi di ottimizzazione (in particolare su reti);</li> <li>3. sulla programmazione lineare e sue applicazioni;</li> <li>4. sull'uso di software di modellazione ed ottimizzazione.</li> </ol>				
Conoscenza e comprensione:				
Sviluppare				
<ol style="list-style-type: none"> <li>1. Capacità di sintesi, di ragionamento logico e di problem solving, mediante tecniche quantitative;</li> <li>2. capacità di formulare e risolvere (anche in maniera approssimata) problemi computazionalmente difficili;</li> <li>3. capacità di usare bibliografia e software in inglese;</li> <li>4. capacità di identificare problemi di programmazione matematica e ottimizzazione;</li> <li>5. capacità di realizzare un modello di un problema matematico, determinando se è lineare, intero, non lineare, di trovare soluzioni per il modello usando gli algoritmi appropriati, di interpretare la soluzione;</li> </ol>				
Applicazione di conoscenza e comprensione:				
<ol style="list-style-type: none"> <li>1. Applicazioni pratiche dei principali problemi di ottimizzazione (in particolare su reti);</li> <li>2. uso di software di modellazione ed ottimizzazione.</li> </ol>				
Autonomia di giudizio:				
Viene rafforzata la autonomia di giudizio dello studente attraverso l'approfondimento della capacità di sintesi, di ragionamento logico-matematico e di problem solving, mediante tecniche quantitative;				
Abilità comunicative:				
Il corso sviluppa le abilità comunicative degli studenti attraverso lavoro in equipe degli studenti finalizzato a simulare la risoluzione di problemi di ottimizzazione che si possono presentare nel contesto reale di un'azienda di logistica.				
Capacità di apprendimento				
Le capacità acquisite nel corso sono di utilità in insegnamenti di livello superiore su temi avanzati nella teoria della complessità, di algoritmi di rete e di teoria dei grafi.				
1022268   SICUREZZA	3°	2°	6	ITA

Insegnamento	Anno	Semestre	CFU	Lingua
<b>Obiettivi formativi</b>				
Obiettivi generali: Il corso di Sicurezza ha l'obiettivo di formare figure professionali capaci di affrontare con successo le sfide costituite dai problemi di sicurezza della società dell'informazione.				
Obiettivi specifici: Il corso prevede lo studio di vari modelli di controllo degli accessi, dell'analisi delle principali difficoltà e risoluzioni di problemi di crittografia e dei principali protocolli di sicurezza utilizzati in rete				
Conoscenza e comprensione: Al superamento dell'esame, lo studente avrà conoscenza e capacità di comprensione delle basi della sicurezza informatica e delle principali tecnologie per l'analisi e la soluzione di problemi di sicurezza.				
Applicare conoscenza e comprensione: Il corso mette in grado lo studente di applicare la propria conoscenza e capacità di comprensione per risolvere problemi di sicurezza informatica, con sufficiente autonomia per affrontare problemi complessi; e per la consultazione efficace di documentazione avanzata di tipo scientifico e tecnologico.				
Autonomia di giudizio: Il corso mira ad acquisire capacità d'interpretazione autonoma per proporre soluzioni a problemi di sicurezza congruenti con le tecnologie disponibili, e di aggiornamento continuo dell'evoluzione tecnologica, per formulare giudizi critici autonomi contribuendo all'avanzamento della sicurezza del sistema.				
Abilità comunicative: Lo studente acquisisce la capacità di presentare e di argomentare le proprie idee in merito ai problemi di sicurezza affrontati ed alle soluzioni proposte, sia con colleghi che con utenti				
Capacità di apprendimento successivo: Il corso prevede lo sviluppo di capacità di approfondimento nell'ambito della sicurezza informatica sia degli aspetti metodologici sia di quelli tecnologici, per adeguarsi al progredire delle tecniche e delle soluzioni ai problemi di sicurezza più comuni, e per proseguire anche in autonomia alla soluzione di nuovi problemi di sicurezza.				
10589652   TECNICHE DI PROGRAMMAZION E FUNZIONALE E IMPERATIVA	3°	2°	6	ITA

Insegnamento	Anno	Semestre	CFU	Lingua
<p><b>Obiettivi formativi</b></p> <p>Obiettivi generali:            Apprendimento del paradigma di programmazione funzionale e di aspetti avanzati di programmazione imperativa.            Studio comparato di diversi paradigmi di programmazione (imperativo, funzionale e a oggetti) per sviluppare un senso critico nell'apprendimento e valutazione dei linguaggi di programmazione.</p> <p>Obiettivi specifici:</p> <p>Conoscenza e Comprensione:            Aspetti avanzati di programmazione C (in particolare uso di puntatori espliciti e (de)allocazione dinamica di memoria, valutazione della complessità concreta dei programmi). Aspetti base ed avanzati di programmazione funzionale in Haskell (in particolare tipi polimorfi, funzionali di ordine superiore, laziness).</p> <p>Applicare Conoscenza e Comprensione:            Applicare la metodologie specifiche di diversi paradigmi di programmazione alla soluzione dei problemi.</p> <p>Capacità Critiche e di Giudizio:            Lo studio comparato dei paradigmi di programmazione permette di valutare ad esempio quale sia il linguaggio più adatto per risolvere un certo problema o sviluppare un progetto software.            Le tecniche di ragionamento sui programmi possono guidare sia il testing che lo sviluppo dei programmi.            Inoltre, lo studente dovrebbe acquisire strumenti necessari per valutare criticamente altri linguaggi di programmazione.</p> <p>Capacità Comunicative            Lo studente è stimolato, soprattutto nella presentazione da fare a fine corso, a esporre concisamente ma in modo preciso le idee necessarie allo sviluppo di una brillante soluzione a un problema computazionale ed anche l'analisi di un programma.</p> <p>Capacità di Apprendimento:            Lo studio comparato di diversi paradigmi di programmazione stimola la capacità di apprendimento di altri linguaggi di programmazione, dando allo studente gli strumenti per distinguere gli aspetti caratterizzanti di un linguaggio di programmazione (logica del core language, come ad esempio passaggio di parametri, sistema dei tipi, semantica dei comandi) da quelli più strettamente tecnici (come imparare a usare librerie e strumenti di sviluppo, come IDE e debugger).</p>				
<p>1021828              MATEMATICA            DISCRETA</p>	3°	2°	6	ITA
<p><b>Obiettivi formativi</b></p> <p>Obiettivi general            L'insegnamento si propone di fornire allo studente un'introduzione alla matematica discreta, che costituisce uno dei settori più innovativi della matematica. Sviluppato a partire dalla seconda metà del novecento, è ricco di problemi stimolanti e di grande utilità nelle applicazioni.</p> <p>Obiettivi specifici            Durante il corso, lo studente verrà a contatto con una serie di argomenti e problemi, di tipo completamente diverso da quelli incontrati in altri corsi di matematica tradizionali, e svilupperà, attraverso un impegno sistematico rivolto al "problem solving", un approccio concreto allo studio di problemi di grande valenza formativa, soprattutto per la futura attività professionale.</p> <p>Conoscenza e comprensione:            Al termine del corso lo studente</p> <ul style="list-style-type: none"> <li>• conoscerà i metodi, i problemi, e le possibili applicazioni della matematica discreta.</li> <li>• sarà in grado di capire, affrontare e risolvere semplici problemi di matematica discreta.</li> <li>• attraverso esercitazioni scritte e eventuali presentazioni orali svilupperà adeguate capacità critiche</li> <li>• allo stesso modo eserciterà la sua capacità di esporre e trasmettere ciò che ha appreso</li> <li>• lo studio individuale allenerà adeguatamente la sua capacità di studio autonomo e indipendente</li> </ul>				

Lo studente deve acquisire 12 CFU fra i seguenti esami

Insegnamento	Anno	Semestre	CFU	Lingua
--------------	------	----------	-----	--------

1022264 |  
LINGUAGGI DI  
PROGRAMMAZION  
E

3°

1°

6

ITA

### Obiettivi formativi

#### Obiettivi generali:

Il corso illustra i fondamenti dei linguaggi di programmazione. Particolare attenzione viene dedicata alla semantica statica e dinamica dei linguaggi (teoria dei tipi, semantica operativa ed assiomatica) ed agli strumenti formali per la specifica e la verifica del software.

#### Obiettivi specifici:

Approfondimento sulle algebre induttive come fondamento di sintassi astratta, strutture dati e semplici logiche (equazionali) di programmi. Cenni sulla co-induzione come fondamento di analisi di processi concorrenti e strutture lazy. Confronto fra linguaggi lazy e eager, statici e dinamici, nel contesto di diversi paradigmi di programmazione: imperativo, funzionale, ad oggetti. Teoria dei tipi, con particolare attenzione al polimorfismo parametrico (alla ML) e di sottoclasse (alla Java).

#### Conoscenze e comprensione:

Alla fine del corso lo studente avrà acquisito conoscenze fondazionali relative ai linguaggi di programmazione e ai principali paradigmi di programmazione, e sarà in grado di inquadrare in una mappa concettuale coerente alcuni fra le principali aree di studio dell'informatica teorica, comprendendone le interconnessioni.

#### Applicazione di conoscenza e comprensione:

Alla fine del corso lo studente sarà in grado di orientarsi fra i diversi paradigmi della programmazione, con particolare attenzione alla programmazione funzionale. Avrà acquisito una conoscenza di base del linguaggio funzionale SML e della sua teoria dei tipi, nonché dell'uso dei metodi formali.

#### Capacità di giudizio:

Gli studenti acquisiranno strumenti culturali per sviluppare software di qualità e per comprendere ed inquadrare le tendenze emergenti dei linguaggi moderni.

#### Capacità di comunicazione:

Gli studenti impareranno ad usare il linguaggio della matematica e della logica per descrivere, sviluppare ed analizzare il software, e ad integrare l'approccio formale con un'esposizione intuitiva delle idee.

#### Capacità di apprendimento:

L'uso di strumenti matematici nella teoria e nella pratica della programmazione sviluppa capacità logiche e consapevolezza della complessità della materia di studio.

10593235 |  
PROGRAMMAZION  
E DI SISTEMI  
EMBEDDED E  
MULTICORE

3°

1°

6

ITA

## Obiettivi formativi

### Obiettivi generali:

Il corso è finalizzato all'acquisizione della conoscenza di tecniche algoritmiche e di programmazione adatte alle moderne piattaforme di calcolo multicore ed embedded.

### Obiettivi specifici:

#### Conoscenza e comprensione:

Al termine del corso gli studenti avranno una profonda comprensione dei principali problemi di programmazione posti dai sistemi embedded e multi-core, nonché delle tecniche di programmazione utili per scrivere codice efficiente e scalabile. Avranno una comprensione della programmazione a basso livello e della progettazione HW.

#### Capacità di applicare conoscenza e comprensione:

gli studenti saranno in grado di progettare, programmare e realizzare sistemi embedded ed avranno una comprensione dei sistemi operativi adottati in tali tipi di sistemi.

#### Capacità critiche e di giudizio:

gli studenti saranno in grado di affrontare le sfide che si presentano nella progettazione di sistemi embedded e multicore e nell'implementazione di programmi efficienti e scalabili per tali sistemi, anche tenendo conto delle caratteristiche architetturali delle diverse piattaforme di calcolo.

#### Capacità di comunicare quanto si è appreso:

gli studenti saranno in grado di comunicare in modo efficace, riassumendo le idee principali nella progettazione di software embedded e multicore in modo chiaro e presentando informazioni tecniche accurate.

#### Capacità di studio autonomo:

obiettivo del corso è di introdurre i principi di base nella progettazione di software embedded e multicore. Gli studenti saranno quindi in grado di estendere la loro conoscenza in modo indipendente, adattandola all'evoluzione delle tecnologie e delle piattaforme di calcolo.

1022262 |  
INTELLIGENZA  
ARTIFICIALE

3°

2°

6

ITA

## Obiettivi formativi

### Obiettivi generali:

Il corso si propone di esporre gli studenti ad un'ampia panoramica dell'Intelligenza Artificiale (IA).

### Obiettivi specifici:

Il corso si propone di mettere gli studenti in grado di comprendere la teoria di un'ampia gamma di tecniche di IA, e di padroneggiarle nella pratica della progettazione di sistemi software intelligenti.

#### Conoscenza e comprensione:

Introduzione ad ampio spettro dei principi fondamentali e delle diverse branche dell'Intelligenza Artificiale (IA) e conoscenze su risoluzione di problemi mediante ricerca, inferenza logica, pianificazione, ragionamento, apprendimento.

#### Applicare conoscenza e comprensione:

Saper applicare il portafoglio di tecniche e gli approcci illustrati per la progettazione e realizzazione di sistemi software intelligenti.

#### Capacità critiche e di giudizio:

Gli studenti saranno in grado di prendere autonomamente decisioni razionali sulle tecniche di IA migliori da impiegare nella progettazione di sistemi software intelligenti.

#### Capacità comunicative:

Gli studenti saranno in grado di interagire in modo proficuo con altri ricercatori in IA su un'ampia gamma di argomenti.

#### Capacità di apprendimento:

Gli studenti saranno in grado di ampliare le loro conoscenze in modo autonomo consultando, secondo necessità, la letteratura scientifica sull'IA.

1022265 | MODELLI  
E OTTIMIZZAZIONE

3°

2°

6

ITA

## Obiettivi formativi

### Obiettivi generali:

Il corso introduce lo studente alla modellazione matematica di problemi di ottimizzazione, allo studio della programmazione lineare e intera e alle loro applicazioni in contesti reali.

### Obiettivi specifici:

#### Apprendere:

1. Le principali tecniche di modellazione di problemi usando strumenti logico/matematici;
2. Le proprietà teoriche e sulle applicazioni pratiche dei principali problemi di ottimizzazione (in particolare su reti);
3. sulla programmazione lineare e sue applicazioni;
4. sull'uso di software di modellazione ed ottimizzazione.

#### Conoscenza e comprensione:

##### Sviluppare

1. Capacità di sintesi, di ragionamento logico e di problem solving, mediante tecniche quantitative;
2. capacità di formulare e risolvere (anche in maniera approssimata) problemi computazionalmente difficili;
3. capacità di usare bibliografia e software in inglese;
4. capacità di identificare problemi di programmazione matematica e ottimizzazione;
5. capacità di realizzare un modello di un problema matematico, determinando se è lineare, intero, non lineare, di trovare soluzioni per il modello usando gli algoritmi appropriati, di interpretare la soluzione;

#### Applicazione di conoscenza e comprensione:

1. Applicazioni pratiche dei principali problemi di ottimizzazione (in particolare su reti);
2. uso di software di modellazione ed ottimizzazione.

#### Autonomia di giudizio:

Viene rafforzata la autonomia di giudizio dello studente attraverso l'approfondimento della capacità di sintesi, di ragionamento logico-matematico e di problem solving, mediante tecniche quantitative;

#### Abilità comunicative:

Il corso sviluppa le abilità comunicative degli studenti attraverso lavoro in equipe degli studenti finalizzato a simulare la risoluzione di problemi di ottimizzazione che si possono presentare nel contesto reale di un'azienda di logistica.

#### Capacità di apprendimento

Le capacità acquisite nel corso sono di utilità in insegnamenti di livello superiore su temi avanzati nella teoria della complessità, di algoritmi di rete e di teoria dei grafi.

## Obiettivi formativi

### Obiettivi generali:

Il corso di Sicurezza ha l'obiettivo di formare figure professionali capaci di affrontare con successo le sfide costituite dai problemi di sicurezza della società dell'informazione.

### Obiettivi specifici:

Il corso prevede lo studio di vari modelli di controllo degli accessi, dell'analisi delle principali difficoltà e risoluzioni di problemi di crittografia e dei principali protocolli di sicurezza utilizzati in rete

### Conoscenza e comprensione:

Al superamento dell'esame, lo studente avrà conoscenza e capacità di comprensione delle basi della sicurezza informatica e delle principali tecnologie per l'analisi e la soluzione di problemi di sicurezza.

### Applicare conoscenza e comprensione:

Il corso mette in grado lo studente di applicare la propria conoscenza e capacità di comprensione per risolvere problemi di sicurezza informatica, con sufficiente autonomia per affrontare problemi complessi; e per la consultazione efficace di documentazione avanzata di tipo scientifico e tecnologico.

### Autonomia di giudizio:

Il corso mira ad acquisire capacità d'interpretazione autonoma per proporre soluzioni a problemi di sicurezza congruenti con le tecnologie disponibili, e di aggiornamento continuo dell'evoluzione tecnologica, per formulare giudizi critici autonomi contribuendo all'avanzamento della sicurezza del sistema.

### Abilità comunicative:

Lo studente acquisisce la capacità di presentare e di argomentare le proprie idee in merito ai problemi di sicurezza affrontati ed alle soluzioni proposte, sia con colleghi che con utenti

### Capacità di apprendimento successivo:

Il corso prevede lo sviluppo di capacità di approfondimento nell'ambito della sicurezza informatica sia degli aspetti metodologici sia di quelli tecnologici, per adeguarsi al progredire delle tecniche e delle soluzioni ai problemi di sicurezza più comuni, e per proseguire anche in autonomia alla soluzione di nuovi problemi di sicurezza.

10589652 |  
TECNICHE DI  
PROGRAMMAZION  
E FUNZIONALE E  
IMPERATIVA

3°

2°

6

ITA

## Obiettivi formativi

### Obiettivi generali:

Apprendimento del paradigma di programmazione funzionale e di aspetti avanzati di programmazione imperativa. Studio comparato di diversi paradigmi di programmazione (imperativo, funzionale e a oggetti) per sviluppare un senso critico nell'apprendimento e valutazione dei linguaggi di programmazione.

### Obiettivi specifici:

#### Conoscenza e Comprensione:

Aspetti avanzati di programmazione C (in particolare uso di puntatori espliciti e (de)allocazione dinamica di memoria, valutazione della complessità concreta dei programmi). Aspetti base ed avanzati di programmazione funzionale in Haskell (in particolare tipi polimorfi, funzionali di ordine superiore, laziness).

#### Applicare Conoscenza e Comprensione:

Applicare la metodologie specifiche di diversi paradigmi di programmazione alla soluzione dei problemi.

#### Capacità Critiche e di Giudizio:

Lo studio comparato dei paradigmi di programmazione permette di valutare ad esempio quale sia il linguaggio più adatto per risolvere un certo problema o sviluppare un progetto software.

Le tecniche di ragionamento sui programmi possono guidare sia il testing che lo sviluppo dei programmi.

Inoltre, lo studente dovrebbe acquisire strumenti necessari per valutare criticamente altri linguaggi di programmazione.

#### Capacità Comunicative

Lo studente è stimolato, soprattutto nella presentazione da fare a fine corso, a esporre concisamente ma in modo preciso le idee necessarie allo sviluppo di una brillante soluzione a un problema computazionale ed anche l'analisi di un programma.

#### Capacità di Apprendimento:

Lo studio comparato di diversi paradigmi di programmazione stimola la capacità di apprendimento di altri linguaggi di programmazione, dando allo studente gli strumenti per distinguere gli aspetti caratterizzanti di un linguaggio di programmazione (logica del core language, come ad esempio passaggio di parametri, sistema dei tipi, semantica dei comandi) da quelli più strettamente tecnici (come imparare a usare librerie e strumenti di sviluppo, come IDE e debugger).

1047635 |  
MACHINE  
LEARNING

3°

2°

6

ITA

Lo studente deve acquisire 6 CFU fra i seguenti esami

Insegnamento	Anno	Semestre	CFU	Lingua
1022264   LINGUAGGI DI PROGRAMMAZION E	3°	1°	6	ITA

Insegnamento	Anno	Semestre	CFU	Lingua
<b>Obiettivi formativi</b>				
<p><b>Obiettivi generali:</b>            Il corso illustra i fondamenti dei linguaggi di programmazione. Particolare attenzione viene dedicata alla semantica statica e dinamica dei linguaggi (teoria dei tipi, semantica operativa ed assiomatica) ed agli strumenti formali per la specifica e la verifica del software.</p> <p><b>Obiettivi specifici:</b>            Approfondimento sulle algebre induttive come fondamento di sintassi astratta, strutture dati e semplici logiche (equazionali) di programmi. Cenni sulla co-induzione come fondamento di analisi di processi concorrenti e strutture lazy. Confronto fra linguaggi lazy e eager, statici e dinamici, nel contesto di diversi paradigmi di programmazione: imperativo, funzionale, ad oggetti. Teoria dei tipi, con particolare attenzione al polimorfismo parametrico (alla ML) e di sottoclasse (alla Java).</p> <p><b>Conoscenze e comprensione:</b>            Alla fine del corso lo studente avrà acquisito conoscenze fondazionali relative ai linguaggi di programmazione e ai principali paradigmi di programmazione, e sarà in grado di inquadrare in una mappa concettuale coerente alcuni fra le principali aree di studio dell'informatica teorica, comprendendone le interconnessioni.</p> <p><b>Applicazione di conoscenza e comprensione:</b>            Alla fine del corso lo studente sarà in grado di orientarsi fra i diversi paradigmi della programmazione, con particolare attenzione alla programmazione funzionale. Avrà acquisito una conoscenza di base del linguaggio funzionale SML e della sua teoria dei tipi, nonché dell'uso dei metodi formali.</p> <p><b>Capacità di giudizio:</b>            Gli studenti acquisiranno strumenti culturali per sviluppare software di qualità e per comprendere ed inquadrare le tendenze emergenti dei linguaggi moderni.</p> <p><b>Capacità di comunicazione:</b>            Gli studenti impareranno ad usare il linguaggio della matematica e della logica per descrivere, sviluppare ed analizzare il software, e ad integrare l'approccio formale con un'esposizione intuitiva delle idee.</p> <p><b>Capacità di apprendimento:</b>            L'uso di strumenti matematici nella teoria e nella pratica della programmazione sviluppa capacità logiche e consapevolezza della complessità della materia di studio.</p>				
10593235   PROGRAMMAZION E DI SISTEMI EMBEDDED E MULTICORE	3°	1°	6	ITA

**Insegnamento****Anno****Semestre****CFU****Lingua****Obiettivi formativi**

## Obiettivi generali:

Il corso è finalizzato all'acquisizione della conoscenza di tecniche algoritmiche e di programmazione adatte alle moderne piattaforme di calcolo multicore ed embedded.

## Obiettivi specifici:

## Conoscenza e comprensione:

Al termine del corso gli studenti avranno una profonda comprensione dei principali problemi di programmazione posti dai sistemi embedded e multi-core, nonché delle tecniche di programmazione utili per scrivere codice efficiente e scalabile. Avranno una comprensione della programmazione a basso livello e della progettazione HW.

## Capacità di applicare conoscenza e comprensione:

gli studenti saranno in grado di progettare, programmare e realizzare sistemi embedded ed avranno una comprensione dei sistemi operativi adottati in tali tipi di sistemi.

## Capacità critiche e di giudizio:

gli studenti saranno in grado di affrontare le sfide che si presentano nella progettazione di sistemi embedded e multicore e nell'implementazione di programmi efficienti e scalabili per tali sistemi, anche tenendo conto delle caratteristiche architetturali delle diverse piattaforme di calcolo.

## Capacità di comunicare quanto si è appreso:

gli studenti saranno in grado di comunicare in modo efficace, riassumendo le idee principali nella progettazione di software embedded e multicore in modo chiaro e presentando informazioni tecniche accurate.

## Capacità di studio autonomo:

obiettivo del corso è di introdurre i principi di base nella progettazione di software embedded e multicore. Gli studenti saranno quindi in grado di estendere la loro conoscenza in modo indipendente, adattandola all'evoluzione delle tecnologie e delle piattaforme di calcolo.

1022262 |  
INTELLIGENZA  
ARTIFICIALE

3°

2°

6

ITA

**Obiettivi formativi**

## Obiettivi generali:

Il corso si propone di esporre gli studenti ad un'ampia panoramica dell'Intelligenza Artificiale (IA).

## Obiettivi specifici:

Il corso si propone di mettere gli studenti in grado di comprendere la teoria di un'ampia gamma di tecniche di IA, e di padroneggiarle nella pratica della progettazione di sistemi software intelligenti.

## Conoscenza e comprensione:

Introduzione ad ampio spettro dei principi fondamentali e delle diverse branche dell'Intelligenza Artificiale (IA) e conoscenze su risoluzione di problemi mediante ricerca, inferenza logica, pianificazione, ragionamento, apprendimento.

## Applicare conoscenza e comprensione:

Saper applicare il portafoglio di tecniche e gli approcci illustrati per la progettazione e realizzazione di sistemi software intelligenti.

## Capacità critiche e di giudizio:

Gli studenti saranno in grado di prendere autonomamente decisioni razionali sulle tecniche di IA migliori da impiegare nella progettazione di sistemi software intelligenti.

## Capacità comunicative:

Gli studenti saranno in grado di interagire in modo proficuo con altri ricercatori in IA su un'ampia gamma di argomenti.

## Capacità di apprendimento:

Gli studenti saranno in grado di ampliare le loro conoscenze in modo autonomo consultando, secondo necessità, la letteratura scientifica sull'IA.

Insegnamento	Anno	Semestre	CFU	Lingua
1022265   MODELLI E OTTIMIZZAZIONE	3°	2°	6	ITA

#### Obiettivi formativi

##### Obiettivi generali:

Il corso introduce lo studente alla modellazione matematica di problemi di ottimizzazione, allo studio della programmazione lineare e intera e alle loro applicazioni in contesti reali.

##### Obiettivi specifici:

###### Apprendere:

1. Le principali tecniche di modellazione di problemi usando strumenti logico/matematici;
2. Le proprietà teoriche e sulle applicazioni pratiche dei principali problemi di ottimizzazione (in particolare su reti);
3. sulla programmazione lineare e sue applicazioni;
4. sull'uso di software di modellazione ed ottimizzazione.

##### Conoscenza e comprensione:

###### Sviluppare

1. Capacità di sintesi, di ragionamento logico e di problem solving, mediante tecniche quantitative;
2. capacità di formulare e risolvere (anche in maniera approssimata) problemi computazionalmente difficili;
3. capacità di usare bibliografia e software in inglese;
4. capacità di identificare problemi di programmazione matematica e ottimizzazione;
5. capacità di realizzare un modello di un problema matematico, determinando se è lineare, intero, non lineare, di trovare soluzioni per il modello usando gli algoritmi appropriati, di interpretare la soluzione;

##### Applicazione di conoscenza e comprensione:

1. Applicazioni pratiche dei principali problemi di ottimizzazione (in particolare su reti);
2. uso di software di modellazione ed ottimizzazione.

##### Autonomia di giudizio:

Viene rafforzata la autonomia di giudizio dello studente attraverso l'approfondimento della capacità di sintesi, di ragionamento logico-matematico e di problem solving, mediante tecniche quantitative;

##### Abilità comunicative:

Il corso sviluppa le abilità comunicative degli studenti attraverso lavoro in equipe degli studenti finalizzato a simulare la risoluzione di problemi di ottimizzazione che si possono presentare nel contesto reale di un'azienda di logistica.

##### Capacità di apprendimento

Le capacità acquisite nel corso sono di utilità in insegnamenti di livello superiore su temi avanzati nella teoria della complessità, di algoritmi di rete e di teoria dei grafi.

1022268   SICUREZZA	3°	2°	6	ITA
---------------------	----	----	---	-----

Insegnamento	Anno	Semestre	CFU	Lingua
<b>Obiettivi formativi</b>				
<p><b>Obiettivi generali:</b>            Il corso di Sicurezza ha l'obiettivo di formare figure professionali capaci di affrontare con successo le sfide costituite dai problemi di sicurezza della società dell'informazione.</p> <p><b>Obiettivi specifici:</b>            Il corso prevede lo studio di vari modelli di controllo degli accessi, dell'analisi delle principali difficoltà e risoluzioni di problemi di crittografia e dei principali protocolli di sicurezza utilizzati in rete</p> <p><b>Conoscenza e comprensione:</b>            Al superamento dell'esame, lo studente avrà conoscenza e capacità di comprensione delle basi della sicurezza informatica e delle principali tecnologie per l'analisi e la soluzione di problemi di sicurezza.</p> <p><b>Applicare conoscenza e comprensione:</b>            Il corso mette in grado lo studente di applicare la propria conoscenza e capacità di comprensione per risolvere problemi di sicurezza informatica, con sufficiente autonomia per affrontare problemi complessi; e per la consultazione efficace di documentazione avanzata di tipo scientifico e tecnologico.</p> <p><b>Autonomia di giudizio:</b>            Il corso mira ad acquisire capacità d'interpretazione autonoma per proporre soluzioni a problemi di sicurezza congruenti con le tecnologie disponibili, e di aggiornamento continuo dell'evoluzione tecnologica, per formulare giudizi critici autonomi contribuendo all'avanzamento della sicurezza del sistema.</p> <p><b>Abilità comunicative:</b>            Lo studente acquisisce la capacità di presentare e di argomentare le proprie idee in merito ai problemi di sicurezza affrontati ed alle soluzioni proposte, sia con colleghi che con utenti</p> <p><b>Capacità di apprendimento successivo:</b>            Il corso prevede lo sviluppo di capacità di approfondimento nell'ambito della sicurezza informatica sia degli aspetti metodologici sia di quelli tecnologici, per adeguarsi al progredire delle tecniche e delle soluzioni ai problemi di sicurezza più comuni, e per proseguire anche in autonomia alla soluzione di nuovi problemi di sicurezza.</p>				
10589652   TECNICHE DI PROGRAMMAZION E FUNZIONALE E IMPERATIVA	3°	2°	6	ITA

Insegnamento	Anno	Semestre	CFU	Lingua
<b>Obiettivi formativi</b>				
Obiettivi generali: Apprendimento del paradigma di programmazione funzionale e di aspetti avanzati di programmazione imperativa. Studio comparato di diversi paradigmi di programmazione (imperativo, funzionale e a oggetti) per sviluppare un senso critico nell'apprendimento e valutazione dei linguaggi di programmazione.				
Obiettivi specifici:				
Conoscenza e Comprensione: Aspetti avanzati di programmazione C (in particolare uso di puntatori espliciti e (de)allocazione dinamica di memoria, valutazione della complessità concreta dei programmi). Aspetti base ed avanzati di programmazione funzionale in Haskell (in particolare tipi polimorfi, funzionali di ordine superiore, laziness).				
Applicare Conoscenza e Comprensione: Applicare la metodologie specifiche di diversi paradigmi di programmazione alla soluzione dei problemi.				
Capacità Critiche e di Giudizio: Lo studio comparato dei paradigmi di programmazione permette di valutare ad esempio quale sia il linguaggio più adatto per risolvere un certo problema o sviluppare un progetto software. Le tecniche di ragionamento sui programmi possono guidare sia il testing che lo sviluppo dei programmi. Inoltre, lo studente dovrebbe acquisire strumenti necessari per valutare criticamente altri linguaggi di programmazione.				
Capacità Comunicative Lo studente è stimolato, soprattutto nella presentazione da fare a fine corso, a esporre concisamente ma in modo preciso le idee necessarie allo sviluppo di una brillante soluzione a un problema computazionale ed anche l'analisi di un programma.				
Capacità di Apprendimento: Lo studio comparato di diversi paradigmi di programmazione stimola la capacità di apprendimento di altri linguaggi di programmazione, dando allo studente gli strumenti per distinguere gli aspetti caratterizzanti di un linguaggio di programmazione (logica del core language, come ad esempio passaggio di parametri, sistema dei tipi, semantica dei comandi) da quelli più strettamente tecnici (come imparare a usare librerie e strumenti di sviluppo, come IDE e debugger).				
1047635   MACHINE LEARNING	3°	2°	6	ENG

Lo studente deve acquisire 6 CFU fra i seguenti esami

Insegnamento	Anno	Semestre	CFU	Lingua
1022263   INTERAZIONE UOMO MACCHINA	3°	1°	6	ITA

Insegnamento	Anno	Semestre	CFU	Lingua
<b>Obiettivi formativi</b>				
Obiettivi generali:				
Al termine del corso gli studenti conoscono le teorie, i modelli e le regole che guidano il progetto e lo sviluppo e la validazione di interfacce e sistemi interattivi usabili.				
Gli studenti che superano l'esame sono in grado di progettare sistemi interattivi seguendo i criteri dell'interazione uomo-computer, analizzando il ruolo dell'utente, gli scenari e i compiti principali, e tenendo in considerazione i vincoli implementativi mediante cicli di progetto e sviluppo molto brevi.				
Obiettivi specifici:				
Conoscenza e comprensione:				
Al termine del corso gli studenti conoscono le teorie, i modelli e le regole che guidano il progetto di interfacce e sistemi interattivi usabili.? Conoscono inoltre i principi di progettazione agile centrata sull'utente.?				
Applicare conoscenza e comprensione:				
?Gli studenti applicano le conoscenze acquisite nella progettazione di un'interfaccia come lavoro di gruppo per l'esame.				
Capacità critiche e di giudizio:				
?Gli studenti, anche attraverso esercitazioni pratiche, acquisiscono competenze nella valutazione e validazione di interfacce uomo computer e ?sviluppano capacità di giudizio sull'usabilità di un'interfaccia e quindi sulle ricadute dell'uso dell'interfaccia in termini di efficacia, efficienza e soddisfazione.				
Capacità comunicative:				
?Gli studenti sostengono due presentazioni del loro lavoro di gruppo in occasione delle due revisioni previste con il docente. La prima revisione è svolta in aula e la presentazione è pertanto rivolta a tutti i colleghi al fine di esercitare le capacità comunicative.?				
Capacità di apprendimento:				
?La capacità di apprendimento? è stimolata attraverso 1) attività di progettazione guidata e autonoma con supervisione; 2) l'esposizione a problemi realistici di progettazione stimolando la ricerca autonoma di soluzioni non standard; 3) la presentazione di casi reali e stimolandone la discussione critica.				
10596283   ORGANIZZAZIONE E GESTIONE PER LO START-UP AZIENDALE	3°	1°	6	ITA

**Obiettivi formativi**

## Obiettivi generali:

L'insegnamento di Organizzazione e Gestione per lo Start-Up Aziendale intende fornire agli studenti le conoscenze e i "fondamentali" per l'analisi e la comprensione delle principali problematiche organizzative che caratterizzano l'avvio e la gestione delle imprese nell'era digitale ed un ampio ventaglio di strumenti utili per la loro risoluzione.

In particolare, una volta affrontati i concetti di base relativi all'impresa e alle dinamiche aziendali, l'insegnamento approfondisce le principali tematiche relative allo start-up, quali la progettazione organizzativa, la pianificazione strategica, la valutazione della business idea e della relativa fattibilità economico-finanziaria.

Tali tematiche vengono affrontate nell'arco di un percorso didattico che si dispiega seguendo cronologicamente gli step funzionali alla redazione di un business plan. Più precisamente, l'insegnamento alterna momenti di lezione frontale ad attività di laboratorio, nelle quali gli studenti, divisi in gruppi, saranno guidati didatticamente nella realizzazione di un business plan relativo ad un'idea d'impresa digitale. Tale percorso consente agli studenti di esaminare in prima persona le caratteristiche del processo di start-up, e li aiuta a comprendere le variabili (e le leve) fondamentali dell'organizzazione, sulle quali la direzione aziendale può agire per il miglioramento delle performance ed il conseguimento di posizioni di vantaggio competitivo sui concorrenti.

Non sono previsti particolari prerequisiti, in termini di conoscenze pregresse di natura economico-aziendale, per la frequenza del corso ed il sostenimento delle prove d'esame.

## Obiettivi specifici

Obiettivi: nella prima parte si descriveranno e si analizzeranno le caratteristiche, gli elementi di base e gli attori che la caratterizzano l'azienda, nonché le fasi del suo ciclo vitale. Si approfondiranno, inoltre, i rapporti esistenti tra l'organizzazione e la gestione dell'azienda, introducendo il concetto di "equilibrio" economico quale legge fondamentale alla base della capacità di sopravvivenza dell'azienda nel tempo.

Obiettivi: nella seconda parte inizierà il percorso didattico atto a fornire le conoscenze necessarie per procedere alla realizzazione del business plan. Si partirà dalla disamina delle opportunità di generazione di start-up nell'era digitale, evidenziandone le differenze con lo start-up di imprese tradizionali. Si esamineranno i fondamenti della strategia, con il passaggio dalla business idea alla formalizzazione del modello di business, i contenuti di base della progettazione organizzativa, con particolare riguardo al rapporto tra organizzazione, ambiente e strategia, le tecniche di analisi della fattibilità di mercato e di quella economico-finanziaria, le considerazioni alla base della scelta della forma giuridica, nonché quelle per l'individuazione delle forme di finanziamento più appropriate.

Obiettivi: nella terza parte si approfondirà il ruolo e la struttura del business plan. Più precisamente, gli studenti, opportunamente divisi in gruppi, verranno didatticamente guidati attraverso il processo di redazione di un business plan. In questa parte, quindi, il business plan verrà considerato come l'obiettivo finale del corso che, oltre a rappresentare un project work valido ai fini della valutazione finale, verrà presentato nell'ambito di un vero e proprio contest alla fine del corso.

Tali obiettivi saranno perseguiti attraverso l'adozione di un appropriato mix di lezioni frontali, testimonianze provenienti dal mondo imprenditoriale e attività laboratoriali di gruppo.

## Conoscenza e capacità di comprensione (knowledge and understanding).

Al termine dell'insegnamento gli studenti saranno in grado di riconoscere ed affrontare analiticamente le problematiche organizzative che caratterizzano la fase dello start-up, potendo contare su un ampio background teorico e un toolkit adeguato alla loro soluzione. Più precisamente, la comprensione dei legami tra variabili ambientali, strategiche ed organizzative consentirà loro di approcciare lo sviluppo dell'idea imprenditoriale ricercando la coerenza degli elementi strategico-organizzativi con il mercato di riferimento, nel rispetto delle condizioni di economicità dell'impresa. Inoltre, grazie all'acquisizione degli strumenti necessari alla progettazione di una start-up digitale e alla conoscenza maturata nel processo di redazione del business plan, alla fine del corso gli studenti saranno in grado di sviluppare un'idea imprenditoriale e di valutarne la fattibilità, nonché di formalizzarla all'interno di un business plan per procedere alla sua concreta realizzazione.

Tali conoscenze saranno acquisite alternando momenti di lezione frontale volti a fornire gli strumenti conoscitivi necessari alla comprensione dei vari argomenti, ad attività laboratoriali di gruppo, nelle quali gli studenti, divisi in gruppi, saranno guidati didatticamente dal docente nella realizzazione di un business plan. Saranno inoltre previste testimonianze provenienti dal mondo imprenditoriale in grado di rappresentare dei casi di studio pratici ed esemplificativi di quanto affrontato dal punto di vista teorico.

## Capacità di applicare conoscenza e comprensione (applying knowledge and understanding).

Attraverso i lavori individuali e di gruppo gli studenti saranno messi in condizione non solo di comprendere l'origine, la natura e la tipologia delle varie problematiche organizzative, ma di elaborare soluzioni condivise utilizzando tecniche, modelli e strumenti di analisi trasferiti durante le lezioni frontali.

La presentazione, da parte degli studenti, dei risultati dei lavori di gruppo consentirà di sviluppare capacità di riflessione e di argomentazione, unitamente a capacità di scambio, condivisione ed esposizione dei risultati conseguiti.

Le testimonianze degli esperti di organizzazione provenienti da primarie organizzazioni imprenditoriali e la discussione dei relativi temi con gli studenti, infine, permetterà di corroborare – anche con un riscontro pratico – la capacità di applicazione delle tecniche e i risultati conseguiti dagli studenti stessi, e di apprendere dagli eventuali errori commessi.

## Autonomia di giudizio (making judgements).

Insegnamento	Anno	Semestre	CFU	Lingua
10606972   DATA SCIENCE E DIRITTO	3°	1°	6	ITA
1031337   BIOLOGIA COMPUTAZIONALE	3°	2°	6	ITA

### Obiettivi formativi

#### Obiettivi generali:

Obiettivo generale del corso è di trasferire agli studenti le conoscenze dello stato dell'arte della biologia computazionale a seguito dell'avvento di tecnologie di sequenziamento massivo per la produzione di dati genomici e proteomici. Tali basi sono necessarie poi per permettere agli studenti di raggiungere le competenze per una corretta analisi delle problematiche proprie dell'area e la capacità di progettazione ed implementazione di un software adatto alla risoluzione del problema proposto. Pertanto il percorso formativo è finalizzato alla formazione di una figura professionale che sia in grado di intervenire nella risoluzione e gestione di progetti informatici in ambito biomolecolare.

#### Obiettivi specifici:

Il corso si propone di formare persone esperte nelle analisi dei dati biomedici e progettisti di sistemi software che possiedano le conoscenze di base della biologia molecolare e degli strumenti bio-informatici utilizzati per affrontare la gestione dell'enorme flusso di dati generati in questo settore. Persone con tale profilo professionale dovranno essere in grado, partendo dalla piattaforma sperimentale di produzione dei dati (saranno particolarmente dettagliate le problematiche derivanti dai dati prodotti con sequenziamento massivo), di stabilire quali siano gli algoritmi di interesse per l'analisi dei dati grezzi del progetto. Dovranno inoltre acquisire una sensibilità critica ed essere in grado di definire un protocollo di analisi dei dati tenendo conto delle risorse di calcolo disponibili e, di conseguenza, ottimizzare l'analisi. Al termine del corso, gli studenti presenteranno anche strumenti da loro sviluppati per la gestione, l'integrazione e l'interrogazione delle enormi quantità di dati prodotti dalle analisi al fine di ottenere risultati finali biologici, efficaci e fruibili. Tali strumenti seguiranno gli standard richiesti nello sviluppo software tipici della comunità bioinformatica.

#### Applicazione di conoscenza e comprensione:

Gli obiettivi formativi sono realizzati attraverso lezioni frontali, attività di laboratorio ed esercitazioni nelle quali sono previste simulazioni di progetti di lavoro, svolgimento in classe o discussione con partecipazione diretta degli studenti relativamente a problemi e all'analisi di casi di studio. Durante le esercitazioni gli studenti apprenderanno come progettare e sviluppare

- una pipeline di analisi bioinformatica per il processamento dei dati grezzi forniti dalle piattaforme di sequenziamento di nuova generazione (NGS)
- l'automatizzazione ed ottimizzazione della pipeline di analisi NGS
- un sistema software per la gestione ed interrogazione dei dati prodotti dall'analisi
- simulazioni di docking e dinamica molecolare di macromolecole biologiche in ambiente di calcolo ad alte prestazioni (HPC).

#### Autonomia di giudizio:

Gli studenti del corso acquisiranno la capacità di elaborare informazioni complesse e/o frammentarie (ad esempio dovranno gestire dati di sequenze annotate solo in parte, ossia solo alcune di esse saranno associate ad un intervallo cromosomico di un organismo sequenziato, e spesso annotate in maniera non standard) e dovranno pervenire ad una modellazione dei dati pensata in maniera originale ed autonoma, scelta coerentemente con l'ambito biologico del proprio progetto sperimentale.

#### Abilità comunicative:

Gli studenti saranno in grado di dialogare con i ricercatori dell'area biomedica, in modo chiaro, logico ed efficace, utilizzando gli strumenti metodologici acquisiti durante il corso e attraverso termini propri della biologia computazionale. L'acquisizione di queste competenze sarà testata attraverso una verifica orale e diversi progetti sviluppati in laboratorio.

#### Capacità di apprendimento:

Gli studenti dovranno aver acquisito la capacità critica, originale ed autonoma di rapportarsi a problematiche proprie dei progetti di biologia computazionale e di applicare autonomamente le conoscenze acquisite durante il corso in vista di un'eventuale prosecuzione degli studi a livello superiore (laurea specialistica) o nella più ampia prospettiva di approfondimento culturale e professionale nel caso di un impiego nell'area biomedica/bioinformatica.

1041761   LINGUAGGI E COMPILATORI	3°	2°	6	ITA
---	----	----	---	-----

Insegnamento	Anno	Semestre	CFU	Lingua
--------------	------	----------	-----	--------

**Obiettivi formativi**

Obiettivi generali:

Lo studente imparerà come funziona un compilatore sia dal punto di vista teorico che pratico.

Obiettivi specifici:

Conoscenza e comprensione:

Allo studente saranno presentate le parti della Teoria dei Linguaggi Formali e degli Automi necessarie ad avere una base teorica sul funzionamento dei compilatori.

Applicare conoscenza e comprensione:

Lo studente imparerà come usare noti strumenti pratici per la costruzione di compilatori, come Lex a YACC.

Capacità critiche e di giudizio:

Una parte facoltativa dell'esame consisterà nel produrre un compilatore per un semplice linguaggio di programmazione ad alto livello.

Capacità comunicative:

Il corso non si propone espliciti obiettivi sulle capacità comunicative, eccetto di formare all'esposizione rigorosa degli argomenti formali.

Capacità di apprendimento:

I fondamenti teorici e gli strumenti pratici introdotti nel corso forniranno allo studente una base solida per studi più avanzati in questo campo.

1022267   PROGRAMMAZION E PER IL WEB	3°	2°	6	ITA
--	----	----	---	-----

**Obiettivi formativi**

Obiettivi generali:

Lo scopo del corso è lo studio del linguaggio java lato server, per la realizzazione di applicazioni web. Il corso fornisce inoltre un'analisi critica di diverse soluzioni implementative di molte funzionalità applicative comuni nello sviluppo di applicazioni web.

Obiettivi specifici:

Programmazione lato server mediante Java Servlet e pagine JSP.

Conoscenza e comprensione:

Il corso permetterà la comprensione del supporto fornito dal linguaggio Java alla realizzazione di applicazioni web. In particolare lo studente studierà le motivazioni alla base di tutte le scelte implementative, in particolare facendo riferimento alle architetture client-server e ai protocolli di rete in uso.

Applicare conoscenza e comprensione:

Attraverso il corso lo studente imparerà a riconoscere tra le possibili soluzioni applicative ad un problema, quale offra i maggiori vantaggi in termini di prestazioni, sicurezza, portabilità e efficienza.

Capacità critiche e di giudizio:

Il corso metterà lo studente in condizione di saper scegliere, dato un problema, la migliore metodologia risolutiva, attraverso la profonda comprensione dei requisiti e dei vincoli imposti dall'architettura.

Capacità comunicative:

Lo studente sarà in grado di motivare le proprie scelte nella proposta di uno specifico approccio realizzativo di un'applicazione Web, e fornire una analisi comparativa dell'approccio scelto con altri approcci possibili.

Capacità di apprendimento:

Lo studente svilupperà capacità di studio autonome e di comprensione e valutazione critica di nuove metodologie, tecnologie e modelli di sviluppo di applicazioni Web.

<b>Insegnamento</b>	<b>Anno</b>	<b>Semestre</b>	<b>CFU</b>	<b>Lingua</b>
10600494   VERIFICA E VALIDAZIONE DI SISTEMI INTELLIGENTI	3°	2°	6	ITA

#### **Obiettivi formativi**

##### Obiettivi generali:

Il corso si propone di fornire agli studenti gli strumenti di base per la modellazione, analisi e progettazione di sistemi intelligenti.

##### Obiettivi specifici:

Il corso si propone di mettere gli studenti in grado di comprendere ed usare un'ampia gamma di tecniche di modellazione, progettazione, verifica e validazione per sistemi intelligenti.

##### Conoscenza e comprensione:

Introduzione ad ampio spettro dei principi fondamentali della modellazione, analisi e progettazione dei sistemi intelligenti modellati come sistemi dinamici sia tempo continuo che tempo discreto.

##### Applicazione di conoscenza e comprensione:

Saper applicare il portafoglio di tecniche e gli approcci illustrati per la modellazione, progettazione, verifica e validazione di sistemi intelligenti.

##### Autonomia di giudizio:

Gli studenti saranno in grado di prendere autonomamente decisioni razionali sulle tecniche da impiegare nella modellazione, progettazione, verifica e validazione di sistemi intelligenti.

##### Abilità comunicative:

Gli studenti saranno in grado di interagire in modo proficuo con esperti di dominio su un'ampia gamma di argomenti relativi alla modellazione, progettazione, verifica e validazione di sistemi intelligenti.

##### Capacità di apprendimento:

Gli studenti saranno in grado di ampliare le loro conoscenze in modo autonomo consultando, secondo necessità, la letteratura scientifica di rilievo.

Lo studente deve acquisire 6 CFU fra i seguenti esami

<b>Insegnamento</b>	<b>Anno</b>	<b>Semestre</b>	<b>CFU</b>	<b>Lingua</b>
1022263   INTERAZIONE UOMO MACCHINA	3°	1°	6	ITA

Insegnamento	Anno	Semestre	CFU	Lingua
<b>Obiettivi formativi</b>				
Obiettivi generali:				
Al termine del corso gli studenti conoscono le teorie, i modelli e le regole che guidano il progetto e lo sviluppo e la validazione di interfacce e sistemi interattivi usabili.				
Gli studenti che superano l'esame sono in grado di progettare sistemi interattivi seguendo i criteri dell'interazione uomo-computer, analizzando il ruolo dell'utente, gli scenari e i compiti principali, e tenendo in considerazione i vincoli implementativi mediante cicli di progetto e sviluppo molto brevi.				
Obiettivi specifici:				
Conoscenza e comprensione:				
Al termine del corso gli studenti conoscono le teorie, i modelli e le regole che guidano il progetto di interfacce e sistemi interattivi usabili.? Conoscono inoltre i principi di progettazione agile centrata sull'utente.?				
Applicare conoscenza e comprensione:				
?Gli studenti applicano le conoscenze acquisite nella progettazione di un'interfaccia come lavoro di gruppo per l'esame.				
Capacità critiche e di giudizio:				
?Gli studenti, anche attraverso esercitazioni pratiche, acquisiscono competenze nella valutazione e validazione di interfacce uomo computer e ?sviluppano capacità di giudizio sull'usabilità di un'interfaccia e quindi sulle ricadute dell'uso dell'interfaccia in termini di efficacia, efficienza e soddisfazione.				
Capacità comunicative:				
?Gli studenti sostengono due presentazioni del loro lavoro di gruppo in occasione delle due revisioni previste con il docente. La prima revisione è svolta in aula e la presentazione è pertanto rivolta a tutti i colleghi al fine di esercitare le capacità comunicative.?				
Capacità di apprendimento:				
?La capacità di apprendimento? è stimolata attraverso 1) attività di progettazione guidata e autonoma con supervisione; 2) l'esposizione a problemi realistici di progettazione stimolando la ricerca autonoma di soluzioni non standard; 3) la presentazione di casi reali e stimolandone la discussione critica.				
1031337   BIOLOGIA COMPUTAZIONALE	3°	2°	6	ITA

Insegnamento	Anno	Semestre	CFU	Lingua
<p><b>Obiettivi formativi</b></p> <p>Obiettivi generali:            Obiettivo generale del corso è di trasferire agli studenti le conoscenze dello stato dell'arte della biologia computazionale a seguito dell'avvento di tecnologie di sequenziamento massivo per la produzione di dati genomici e proteomici. Tali basi sono necessarie poi per permettere agli studenti di raggiungere le competenze per una corretta analisi delle problematiche proprie dell'area e la capacità di progettazione ed implementazione di un software adatto alla risoluzione del problema proposto. Pertanto il percorso formativo è finalizzato alla formazione di una figura professionale che sia in grado di intervenire nella risoluzione e gestione di progetti informatici in ambito biomolecolare.</p> <p>Obiettivi specifici:            Il corso si propone di formare persone esperte nelle analisi dei dati biomedici e progettisti di sistemi software che possiedano le conoscenze di base della biologia molecolare e degli strumenti bio-informatici utilizzati per affrontare la gestione dell'enorme flusso di dati generati in questo settore. Persone con tale profilo professionale dovranno essere in grado, partendo dalla piattaforma sperimentale di produzione dei dati (saranno particolarmente dettagliate le problematiche derivanti dai dati prodotti con sequenziamento massivo), di stabilire quali siano gli algoritmi di interesse per l'analisi dei dati grezzi del progetto. Dovranno inoltre acquisire una sensibilità critica ed essere in grado di definire un protocollo di analisi dei dati tenendo conto delle risorse di calcolo disponibili e, di conseguenza, ottimizzare l'analisi. Al termine del corso, gli studenti presenteranno anche strumenti da loro sviluppati per la gestione, l'integrazione e l'interrogazione delle enormi quantità di dati prodotti dalle analisi al fine di ottenere risultati finali biologici, efficaci e fruibili. Tali strumenti seguiranno gli standard richiesti nello sviluppo software tipici della comunità bioinformatica.</p> <p>Applicazione di conoscenza e comprensione:            Gli obiettivi formativi sono realizzati attraverso lezioni frontali, attività di laboratorio ed esercitazioni nelle quali sono previste simulazioni di progetti di lavoro, svolgimento in classe o discussione con partecipazione diretta degli studenti relativamente a problemi e all'analisi di casi di studio. Durante le esercitazioni gli studenti apprenderanno come progettare e sviluppare</p> <ul style="list-style-type: none"> <li>• una pipeline di analisi bioinformatica per il processamento dei dati grezzi forniti dalle piattaforme di sequenziamento di nuova generazione (NGS)</li> <li>• l'automatizzazione ed ottimizzazione della pipeline di analisi NGS</li> <li>• un sistema software per la gestione ed interrogazione dei dati prodotti dall'analisi</li> <li>• simulazioni di docking e dinamica molecolare di macromolecole biologiche in ambiente di calcolo ad alte prestazioni (HPC).</li> </ul> <p>Autonomia di giudizio:            Gli studenti del corso acquisiranno la capacità di elaborare informazioni complesse e/o frammentarie (ad esempio dovranno gestire dati di sequenze annotate solo in parte, ossia solo alcune di esse saranno associate ad un intervallo cromosomico di un organismo sequenziato, e spesso annotate in maniera non standard) e dovranno pervenire ad una modellazione dei dati pensata in maniera originale ed autonoma, scelta coerentemente con l'ambito biologico del proprio progetto sperimentale.</p> <p>Abilità comunicative:            Gli studenti saranno in grado di dialogare con i ricercatori dell'area biomedica, in modo chiaro, logico ed efficace, utilizzando gli strumenti metodologici acquisiti durante il corso e attraverso termini propri della biologia computazionale. L'acquisizione di queste competenze sarà testata attraverso una verifica orale e diversi progetti sviluppati in laboratorio.</p> <p>Capacità di apprendimento:            Gli studenti dovranno aver acquisito la capacità critica, originale ed autonoma di rapportarsi a problematiche proprie dei progetti di biologia computazionale e di applicare autonomamente le conoscenze acquisite durante il corso in vista di un'eventuale prosecuzione degli studi a livello superiore (laurea specialistica) o nella più ampia prospettiva di approfondimento culturale e professionale nel caso di un impiego nell'area biomedica/bioinformatica.</p>				
1041761   LINGUAGGI E COMPILATORI	3°	2°	6	ITA

Insegnamento	Anno	Semestre	CFU	Lingua
<b>Obiettivi formativi</b>				
Obiettivi generali: Lo studente imparerà come funziona un compilatore sia dal punto di vista teorico che pratico.				
Obiettivi specifici: Conoscenza e comprensione: Allo studente saranno presentate le parti della Teoria dei Linguaggi Formali e degli Automi necessarie ad avere una base teorica sul funzionamento dei compilatori.				
Applicare conoscenza e comprensione: Lo studente imparerà come usare noti strumenti pratici per la costruzione di compilatori, come Lex a YACC.				
Capacità critiche e di giudizio: Una parte facoltativa dell'esame consisterà nel produrre un compilatore per un semplice linguaggio di programmazione ad alto livello.				
Capacità comunicative: Il corso non si propone espliciti obiettivi sulle capacità comunicative, eccetto di formare all'esposizione rigorosa degli argomenti formali.				
Capacità di apprendimento: I fondamenti teorici e gli strumenti pratici introdotti nel corso forniranno allo studente una base solida per studi più avanzati in questo campo.				
1022267   PROGRAMMAZION E PER IL WEB	3°	2°	6	ITA
<b>Obiettivi formativi</b>				
Obiettivi generali: Lo scopo del corso è lo studio del linguaggio java lato server, per la realizzazione di applicazioni web. Il corso fornisce inoltre un'analisi critica di diverse soluzioni implementative di molte funzionalità applicative comuni nello sviluppo di applicazioni web.				
Obiettivi specifici: Programmazione lato server mediante Java Servlet e pagine JSP.				
Conoscenza e comprensione: Il corso permetterà la comprensione del supporto fornito dal linguaggio Java alla realizzazione di applicazioni web. In particolare lo studente studierà le motivazioni alla base di tutte le scelte implementative, in particolare facendo riferimento alle architetture client-server e ai protocolli di rete in uso.				
Applicare conoscenza e comprensione: Attraverso il corso lo studente imparerà a riconoscere tra le possibili soluzioni applicative ad un problema, quale offra i maggiori vantaggi in termini di prestazioni, sicurezza, portabilità e efficienza.				
Capacità critiche e di giudizio: Il corso metterà lo studente in condizione di saper scegliere, dato un problema, la migliore metodologia risolutiva, attraverso la profonda comprensione dei requisiti e dei vincoli imposti dall'architettura.				
Capacità comunicative: Lo studente sarà in grado di motivare le proprie scelte nella proposta di uno specifico approccio realizzativo di un'applicazione Web, e fornire una analisi comparativa dell'approccio scelto con altri approcci possibili.				
Capacità di apprendimento: Lo studente svilupperà capacità di studio autonome e di comprensione e valutazione critica di nuove metodologie, tecnologie e modelli di sviluppo di applicazioni Web.				

Insegnamento	Anno	Semestre	CFU	Lingua
10600494   VERIFICA E VALIDAZIONE DI SISTEMI INTELLIGENTI	3°	2°	6	ITA

### Obiettivi formativi

#### Obiettivi generali:

Il corso si propone di fornire agli studenti gli strumenti di base per la modellazione, analisi e progettazione di sistemi intelligenti.

#### Obiettivi specifici:

Il corso si propone di mettere gli studenti in grado di comprendere ed usare un'ampia gamma di tecniche di modellazione, progettazione, verifica e validazione per sistemi intelligenti.

#### Conoscenza e comprensione:

Introduzione ad ampio spettro dei principi fondamentali della modellazione, analisi e progettazione dei sistemi intelligenti modellati come sistemi dinamici sia tempo continuo che tempo discreto.

#### Applicazione di conoscenza e comprensione:

Saper applicare il portafoglio di tecniche e gli approcci illustrati per la modellazione, progettazione, verifica e validazione di sistemi intelligenti.

#### Autonomia di giudizio:

Gli studenti saranno in grado di prendere autonomamente decisioni razionali sulle tecniche da impiegare nella modellazione, progettazione, verifica e validazione di sistemi intelligenti.

#### Abilità comunicative:

Gli studenti saranno in grado di interagire in modo proficuo con esperti di dominio su un'ampia gamma di argomenti relativi alla modellazione, progettazione, verifica e validazione di sistemi intelligenti.

#### Capacità di apprendimento:

Gli studenti saranno in grado di ampliare le loro conoscenze in modo autonomo consultando, secondo necessità, la letteratura scientifica di rilievo.

## Obiettivi formativi

Il corso di laurea in Informatica ha l'obiettivo di formare figure professionali capaci di affrontare con successo le sfide costituite dalle crescenti esigenze della società dell'informazione. I laureati in Informatica saranno dotati di una solida preparazione culturale di base, che permetterà loro di mantenersi al passo col progredire delle tecnologie, e di una solida preparazione tecnica, che consentirà loro un rapido inserimento professionale nel settore delle tecnologie dell'informazione e della comunicazione. Inoltre, saranno in grado di accedere ai livelli di studio universitario successivi al primo, nel settore dell'informatica. La preparazione culturale di base permetterà ai laureati in Informatica di avere: - familiarità col metodo scientifico di indagine; - capacità di comprendere e utilizzare strumenti matematici di supporto; - conoscenze metodologiche e competenze di base in un ampio spettro di settori delle scienze e delle tecnologie dell'informazione e della comunicazione; - familiarità con almeno una lingua dell'Unione Europea. Grazie a tali solide basi teoriche, metodologiche e tecnologiche, i laureati in Informatica saranno in grado di: - comprendere l'evoluzione tecnologica e adeguarsi al progredire delle discipline informatiche, anche attraverso la consultazione di documentazione avanzata di tipo scientifico e tecnologico; - possedere buone capacità ed autonomia nella costruzione di modelli indispensabili per la comprensione e la formalizzazione di problemi complessi; - operare nella progettazione, sviluppo e gestione di sistemi software (anche in ambiente di rete), di sistemi informativi, di reti di elaboratori, di soluzioni per la sicurezza dei sistemi di calcolo; - fornire supporto tecnologico ad utilizzatori di sistemi informatici; - inserirsi rapidamente ed efficacemente in una realtà lavorativa, operando sia in gruppo che in autonomia; - comunicare ed argomentare le proprie idee in merito ai problemi affrontati ed alle soluzioni proposte, tanto ad interlocutori specialisti che non specialisti; - accedere al successivo livello di studi costituito dalle Lauree magistrali di area Informatica. I laureati in Informatica saranno in grado di svolgere attività professionale sia in aziende produttrici che in aziende utilizzatrici di sistemi informatici, operanti tanto nel settore pubblico che in quello privato, nei seguenti ambiti occupazionali: - progettazione, organizzazione, manutenzione anche evolutiva e gestione di sistemi software, software applicativo, basi di dati, sistemi informativi, applicazioni di rete client-server, siti Web; - progettazione,

organizzazione, manutenzione anche evolutiva delle componenti di supporto alla sicurezza e alla affidabilità dei sistemi informatici. Il percorso formativo si articola nel modo seguente: 1) nel primo anno (i cui insegnamenti sono tutti obbligatori) viene fornita la preparazione matematica di base e vengono fornite le prime conoscenze fondamentali di Informatica, inoltre è previsto un insegnamento relativo alla lingua inglese; 2) nel secondo anno (i cui insegnamenti sono tutti obbligatori) viene completata la necessaria preparazione matematica e vengono fornite ulteriori conoscenze di Informatica su aree la cui conoscenza è irrinunciabile per l'odierno informatico; 3) nel terzo anno si completa la formazione informatica con due insegnamenti obbligatori nell'area dell'ingegneria del software e degli automi, calcolabilità e complessità, e si offre allo studente la possibilità di scegliere in quale direzione approfondire la propria preparazione: potrà quindi orientarsi verso insegnamenti a carattere più teorico o più applicativo, sempre offrendo una visione delle metodologie necessarie nei relativi campi. Completano il terzo anno i crediti a scelta dello studente, un tirocinio formativo obbligatorio e l'esame di laurea. Il tirocinio formativo è svolto sotto la guida di un responsabile interno alla Facoltà e può essere esterno (svolto presso aziende o enti esterni) o interno (svolto nell'ambito del corso di laurea). In entrambi i casi il tirocinio prevede che allo studente sia proposto un problema del mondo reale, che dovrà risolvere attraverso l'elaborazione di un progetto sviluppato con un approccio professionale. Per tutti gli insegnamenti sono previste attività di laboratorio e/o progettazione o esercitazioni. In particolare, quasi tutti gli insegnamenti di area informatica prevedono attività di laboratorio mentre gli insegnamenti di matematica, o comunque a carattere teorico, prevedono delle esercitazioni. Per quanto riguarda le attività di laboratorio, esse sono essenzialmente rivolte allo sviluppo di semplici programmi negli insegnamenti del primo anno, mentre divengono veri e propri laboratori progettuali negli insegnamenti del secondo e, ancor più, del terzo anno. Il regolamento didattico del corso di laurea definisce, nel rispetto dei limiti normativi, la quota dell'impegno orario complessivo a disposizione dello studente per lo studio personale o per altre attività formative di tipo individuale.

## **Profilo professionale**

### **Profilo**

Tecnologo dell'informatica

### **Funzioni**

Il laureato triennale in Informatica che abbia seguito un curriculum orientato in senso tecnologico è un professionista dotato di solide conoscenze di base e di preparazione tecnica adeguata a un rapido ed efficace inserimento in uno dei settori lavorativi collegati agli insegnamenti tecnologici scelti. È quindi formato per affrontare situazioni concrete di lavoro che possono andare dall'analisi e reingegnerizzazione di sistemi già esistenti allo sviluppo di nuovi sistemi o componenti, anche contribuendo all'estrazione e l'analisi dei loro requisiti. In breve, il curriculum orientato in senso tecnologico mette in grado lo studente di ricoprire qualsiasi funzione, a livello junior, che richieda competenze in diversi campi dell'informatica, in relazione ad attività di: - progettazione, gestione e manutenzione di basi di dati e sistemi informativi in generale; - progettazione e valutazione di interfacce utente; - progettazione, gestione e manutenzione di applicazioni di diversa natura e di servizi per il Web; - progettazione, gestione e manutenzione di software di sistema.

### **Competenze**

Gli approfondimenti di natura tecnologica mirano a fornire le seguenti capacità e competenze professionali: - conoscenza dei principi e delle metodologie per la progettazione, lo sviluppo, l'ottimizzazione e la valutazione di una base di dati; - conoscenza delle problematiche fondamentali, e delle relative soluzioni, che si incontrano nella progettazione di interfacce utente da utilizzarsi sia su elaboratori fissi che portatili, nonché le principali tecniche per la valutazione di usabilità; - conoscenza dei costrutti linguistici principali e delle tecnologie più diffuse, con particolare riferimento ai moderni linguaggi di programmazione orientati ad oggetti, per lo sviluppo di contenuti e servizi per il Web; - conoscenza dei metodi e degli strumenti per la programmazione sequenziale e concorrente, per la comunicazione tra processi e, più in generale, per la programmazione di sistema; - esperienza di lavoro in gruppo. Ognuna di queste competenze, in isolamento o in connessione con le altre, è comunemente richiesta in ambiti lavorativi.

### **Sbocchi lavorativi**

La pervasività dell'Informatica nelle attività economiche è tale che praticamente ormai qualsiasi ambiente di lavoro, sia esso pubblico, privato o autonomo, non può non ricorrere a strumenti informatici per lo sviluppo delle

proprie attività né può fare a meno di interfacciarsi con altre entità pure attraverso strumenti informatici. I laureati in Informatica che abbiano seguito un curriculum orientato in senso tecnologico saranno professionisti capace di inserirsi efficacemente sia all'interno di compagnie di sviluppo software, sia all'interno di reparti IT di organizzazioni che utilizzino estesamente strumenti informatici, sia in società di consulenza informatica, che richiedano sofisticate capacità realizzative in ambito informatico, operanti sia nel settore pubblico che in quello privato, Inoltre, la loro formazione potrà essere completata attraverso un corso di laurea di secondo livello di area Informatica in modo da dare loro accesso alle professioni specialistiche presenti nella categoria ISTAT degli Informatici e telematici. Essi possono accedere, senza ulteriori requisiti, all'esame di stato per l'iscrizione all'Albo degli ingegneri dell'informazione (Albo professionale - sezione B degli ingegneri junior - settore dell'informazione).

# **Frequentare**

## **Laurearsi**

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

# Organizzazione

## Presidente del Corso di studio - Presidente del Consiglio di area didattica

Annalisa Massini

## Tutor del corso

ALESSANDRO CHECCO  
GIUSEPPE PERELLI  
MARIA DE MARSICO  
TONI MANCINI  
ANDREA STERBINI  
IACOPO MASI  
MATTIA SAMORY  
ENRICO TRONCI  
MAURIZIO MANCINI  
TIZIANA CALAMONERI  
ANGELO MONTI

## Manager didattico

## Rappresentanti degli studenti

Roberto Sacchetti  
Giovanni Gianfriddo  
Michele Vincenzo Gentile  
Andrea Tarricone  
Agliaia Norza  
Rokshana Ahmed  
Matteo Collica

## Docenti di riferimento

LUIGI ORSINA  
LORENZO CARLUCCI  
ANNALISA MASSINI  
GAIA MASELLI  
GIANNI FRANCESCHINI  
GABRIELE TOLOMEI  
IVANO SALVO  
GIOVANNA NAPPO  
FEDERICO PELLARIN  
ANDREA TERRACINA  
DANIELE FRIOLO  
GABRIELE VIAGGI  
NADIA ANSINI  
MASSIMO GERMANO  
PAUL JOSEPH WOLLAN  
AZAHARA DE LA TORRE PEDRAZA

## Regolamento del corso

Il percorso formativo del corso di laurea in Informatica si articola su tre anni nel modo seguente. 1) Durante il primo anno viene fornita la preparazione matematica di base e vengono fornite le prime conoscenze fondamentali di Informatica. È inoltre previsto un insegnamento relativo alla lingua inglese. Gli insegnamenti del

primo anno sono tutti obbligatori e sono: • Fondamenti di Programmazione • Metodologie di Programmazione • Introduzione agli Algoritmi • Progettazione di Sistemi Digitali • Architettura Degli Elaboratori • Calcolo Differenziale • Calcolo Integrale • Metodi Matematici per l'Informatica • Lingua Inglese 2) Durante il secondo anno viene completata la preparazione matematica necessaria e vengono fornite ulteriori conoscenze di Informatica su aree la cui conoscenza è irrinunciabile per una laureata o un laureato in Informatica Gli insegnamenti del secondo anno sono tutti obbligatori e sono: • Progettazione di Algoritmi • Sistemi Operativi (I e II modulo) • Reti di Elaboratori • Basi di Dati (I e II modulo) • Algebra • Calcolo delle Probabilità 3) Durante il terzo anno sono previsti due insegnamenti obbligatori che permettono di completare la formazione informatica: • Ingegneria del software • Automi, calcolabilità e complessità Le studentesse e gli studenti hanno poi la possibilità di scegliere in quale direzione approfondire la propria preparazione, orientandosi verso insegnamenti a carattere più teorico o più applicativo, che permettono comunque di apprendere le corrette metodologie di progettazione e analisi necessarie negli specifici campi. Gli insegnamenti da scegliere previsti al terzo anno per il completamento della formazione sono: • Linguaggi di Programmazione • Linguaggi e Compilatori • Intelligenza Artificiale • Machine Learning • Programmazione di Sistemi Embedded e MultiCore • Verifica e Validazione dei Sistemi Intelligenti • Programmazione per il Web • Interazione Uomo Macchina • Sicurezza • Tecniche di Programmazione Funzionale e Imperativa • Matematica Discreta (fruito da Discrete Mathematics) • Modelli e Ottimizzazione • Biologia Computazionale • Organizzazione e Gestione per lo Start-Up Aziendale • Data Science e Diritto • Fisica Completano il terzo anno i crediti a scelta dello studente, massimo 12 CFU, che possono essere scelti anche al di fuori dell'offerta formativa del percorso di studi, purché contribuiscano coerentemente al completamento della formazione. Per tutti gli insegnamenti sono previste attività di laboratorio e/o attività di progettazione oppure esercitazioni. In particolare, molti degli insegnamenti di area informatica prevedono attività di tipo laboratoriale, mentre gli insegnamenti di matematica e gli insegnamenti a carattere teorico, prevedono delle esercitazioni. Per quanto riguarda le attività di laboratorio, esse sono essenzialmente rivolte allo sviluppo di programmi negli insegnamenti del primo anno, mentre divengono veri e propri laboratori progettuali negli insegnamenti del secondo e del terzo anno. Per completare il percorso di studi, le studentesse e gli studenti dovranno svolgere un'attività di tirocinio e sostenere l'esame di laurea. Il tirocinio è svolto sotto la guida di un responsabile interno alla Facoltà e può essere esterno (svolto presso aziende o enti esterni) o interno (svolto nell'ambito del corso di laurea). In entrambi i casi il tirocinio prevede che alla studentessa o allo studente sia proposto un problema del mondo reale, che dovrà risolvere attraverso l'elaborazione di un progetto sviluppato con un approccio professionale. I metodi di accertamento delle conoscenze e delle competenze acquisite per i singoli aspetti della formazione si basa su valutazioni in itinere e/o sugli esami finali dei corsi, nonché attraverso la valutazione della documentazione prodotta per progetti e attività di laboratorio. La verifica della preparazione complessiva acquisita durante il corso di studi avviene attraverso la prova finale, che consiste nella presentazione dell'attività di tirocinio svolta sotto la supervisione di un docente alla commissione valutatrice, che potrà porre domande per accertare la padronanza del candidato rispetto al lavoro di tirocinio svolto.

# Assicurazione qualità

## Consultazioni iniziali con le parti interessate

Il giorno 5 giugno 2008 si è svolta una consultazione formale con rappresentanti del CNIPA e di Assinform, preceduta da numerose riunioni informali avvenute in occasione degli incontri semestrali con le aziende organizzati dal Dipartimento di Informatica (<http://w3.uniroma1.it/dipinfo/incontriaziende.asp>). L'esito della consultazione formale è stato il seguente: 'I presenti considerano favorevolmente la razionalizzazione dell'offerta complessiva che si concretizza attraverso la riduzione del numero dei corsi di laurea da due ad uno, prendono atto che nessun rilievo è stato formulato durante la consultazione formale, ed esprimono quindi parere favorevole all'istituzione del corso di laurea in Informatica in applicazione del D.M. 270/2004 e successivi decreti.' Nell'incontro finale della consultazione a livello di Ateneo del 19 gennaio 2009, considerati i risultati della consultazione telematica che lo ha preceduto, le organizzazioni intervenute hanno valutato favorevolmente la razionalizzazione dell'Offerta Formativa della Sapienza, orientata, oltre che ad una riduzione del numero dei corsi, alla loro diversificazione nelle classi che mostrano un'attrattività elevata e per le quali vi è una copertura di docenti più che adeguata. Inoltre, dopo aver valutato nel dettaglio l'Offerta Formativa delle Facoltà, le organizzazioni stesse hanno espresso parere favorevole all'istituzione dei singoli corsi.

## Consultazioni successive con le parti interessate

Nell'incontro finale della consultazione a livello di Ateneo del 22 gennaio 2010, considerati i risultati della consultazione telematica che lo ha preceduto, le organizzazioni intervenute hanno valutato favorevolmente la nuova Offerta Formativa della Sapienza nel suo complesso. Inoltre, dopo aver valutato nel dettaglio l'Offerta Formativa delle Facoltà che hanno presentato i corsi di nuova istituzione e quelli derivanti dalla trasformazione di corsi già istituiti ai sensi del D.M. 509/1999, tenuto conto delle consultazioni effettuate dalle Facoltà proponenti, le organizzazioni stesse hanno espresso parere favorevole all'istituzione dei singoli corsi di studio, giudicando congrui gli obiettivi formativi specifici dei corsi proposti con l'esigenza formativa presente sul territorio. La consultazione sul progetto formativo per l'a.a. 2016/2017 dei corsi di studio della Facoltà è avvenuta nel modo seguente. Il 10.03.2015 è stato organizzato dalla Presidenza un incontro con le Organizzazioni rappresentative della produzione di beni e servizi e delle professioni per tutti i corsi di laurea della Facoltà. All'incontro erano presenti il Preside, il Vice Preside, il Manager Didattico, i Presidenti di CAD/CdS, e le seguenti organizzazioni rappresentative: 5 Emme informatica (Manager e Responsabile area prodotti), BIC Lazio S.p.A. (Responsabile Staff operativo), Cineca – SCIA (Information and knowledge management services), Exaltech - Impresa Latina (Co-fondatore e Vice Presidente), Ey (Reclutatrice risorse umane), GSE - Gestore servizi energetici (Ingegnere gestionale), IBM (Business Development Executive), INFO EDGE, Istituto Italiano degli Attuari (Segretario Generale), Istituto Nazionale della Previdenza Sociale (Coordinatore Sezione statistico-attuariale), KYDEA - impresa di Latina (CTO & Co-founder), Lait Regione Lazio (Amministratore unico), NS12 (Responsabile Marketing & Comunicazione), NttData (Responsabile HR), Ordine Nazionale degli Attuari (Presidente), SAS (SAS Academic Program manager), Telecom Italia/TIM (Strategy & Innovation- Market & Service Scenario e Risorse umane Senior Consultant). È stata svolta un'indagine di Cesop Communication sulla conoscenza e la percezione che le aziende italiane hanno della formazione erogata dai corsi di studio della Facoltà di Ingegneria dell'informazione, Informatica e Statistica. L'indagine ha avuto come scopo quello di effettuare una prima analisi di sfondo su due aspetti dei corsi della Facoltà: la notorietà dei corsi e la qualità percepita. L'indagine si è svolta attraverso la somministrazione di un questionario on-line con sistema CAWI ipostat-interview (domande standard e batterie con scala a intervalli da 1 a 10). Le aree del questionario riguardavano: - mercato del lavoro - esame dell'offerta formativa dei corsi della Facoltà - competenze (hard skill e soft skill) Il questionario è stato inviato a personale con funzioni decisionali nelle risorse umane di aziende operanti in Italia (addetti HR). Le aziende contattate sono state 3800, ma solo 100 questionari sono stati ritenuti validi. I casi non sono stati pesati rispettando i valori presenti nell'universo statistico di riferimento poiché solamente il 73% ha compilato i dati di base (area geografica e grandezza dell'azienda). Il dato che maggiormente si è riscontrato è stato la mancata conoscenza dell'offerta formativa della Facoltà. Questo elemento ha condizionato fortemente tutta l'indagine. Dei 73 intervistati che hanno indicato le caratteristiche base, 16 addetti provengono da piccole aziende, 15 da medie e 42 da grandi aziende. Dal nord hanno risposto in 38 unità, mentre dal centro 31 e solo 4 dal Sud e Isole. La maggioranza dei rispondenti (25 su 73 dichiaranti i dati di base) è occupato in un'azienda del settore IT (information technology). Le aziende del campione operano nei seguenti settori: Information Tecnology (34.7%), Industriale - manifatturiero - Trasporti (19.4%), Consulenza e revisione aziendale (15.3%), Media e comunicazione (9.7%), Chimico - Farmaceutico - biomedicale (6.9%), Grande distribuzione (5.6%), Istituzioni

pubbliche (5.6%), Finanziario - assicurativo (2.8%). Dalle consultazioni è emerso quanto segue: - Alle organizzazioni presenti il 10.03.2016 è stata fornito un opuscolo della Facoltà nel quale, per ogni corso di laurea e laurea magistrale, oltre ad essere indicato il sito web del corso, erano illustrate le finalità del corso, il percorso formativo e gli sbocchi occupazionali. Tale documentazione è stata inviata in data 17.02.2016. Nel corso dell'incontro sono stati posti i seguenti temi per la valutazione dei fabbisogni formativi e degli sbocchi professionali offerti dai corsi di laurea della Facoltà: • Adeguatezza degli obiettivi formativi e delle denominazioni dei corsi di laurea • Adeguatezza delle figure professionali rispetto alle esigenze del mercato del lavoro • Sbocchi professionali attesi • Suggerimenti sugli obiettivi e sui contenuti dei corsi di studio • Opinioni sulle lauree triennali e sulla durata media dei corsi di laurea • Collaborazioni in Stage/Tirocini/Tesi/Ricerca Tutte le organizzazioni hanno ritenuto validi ed interessanti i temi trattati nei corsi della Facoltà e gli sbocchi occupazionali previsti, consigliando di aumentare però la trasversalità tra i corsi di laurea in parte già presente (nei corsi Interfacoltà e Interdipartimentali). Le organizzazioni hanno poi sottolineato come nell'offerta formativa sono presenti i temi attuali come information technology e data science, big data, internet of things (IoT) e internet of everything (IoE), smart cities, robotica, domotica, cyber security, cognitive computing, social networking, cloud analytics, mobile networking, privacy, open source, open data, open agent, auspicandone però ulteriori approfondimenti e sviluppi anche per le esigenze legate all'introduzione della cittadinanza digitale nella Pubblica amministrazione. Inoltre, è stata sottolineata l'importanza di incrementare la formazione normativa soprattutto nei corsi di area statistico e attuariale. Per quanto riguarda il tema delle soft skills, sono emerse posizioni diverse tra le grandi organizzazioni e quelle medio/piccole. Secondo le prime è necessario aumentarne la presenza anche attraverso la collaborazione con le aziende disponibili a fornire seminari da inserire nell'offerta formativa come CFU. Le piccole e medie organizzazioni, invece, ritengono che sia più importante fornire una forte formazione di base soprattutto nelle lauree di primo livello. Tutte le organizzazioni hanno sottolineato come sia importante, ai fini del collocamento sul mercato del lavoro, che gli studenti conseguano il titolo in corso e che già durante il corso di studi inizino la collaborazione con mondo del lavoro, aumentando la previsione di testimonianze aziendali all'interno dei corsi, l'attivazione di stage, di progetti di ricerca e di collaborazioni nella stesura della tesi. Per attuare questi aspetti le organizzazioni hanno manifestato la loro piena collaborazione. Dalle tre aree del questionario dell'Indagine Cesop Communication è emerso che: - l'88% del campione prevede di assumere nel 2016 avvalendosi di risorse con formazione proveniente dalla Facoltà di Ingegneria dell'Informazione, Informatica e Statistica. L'ingegnere informatico in particolare è la figura professionale che riscuote maggior successo (19.8% su base 88). - La conoscenza dell'offerta formativa dei corsi risulta molto bassa e per questo la comunicazione della Facoltà con le aziende dovrebbe essere incrementata (collaborazioni e incontri tra Facoltà e aziende). Un interessante benchmarking è il Politecnico di Milano che viene considerato un Ateneo di qualità e con una buona comunicazione. - Sul piano dei contenuti dell'offerta formativa le aziende, anche se di poco, premiano per importanza le soft skills rispetto alle hard skills (media 5.58 su scala 10). Tra le soft skills la «Capacità di collaborare con gli altri in modo costruttivo» e la «Capacità di adattamento alle esigenze delle organizzazioni» sono quelle sulle quali la Facoltà deve e può incidere maggiormente. I CdS concordano di tenere conto delle seguenti indicazioni: - continuare ad approfondire nei corsi di laurea e laurea magistrale temi come information technology e data science, big data, Internet of Things (IoT) e Internet of Everything (IoE), smart cities, robotica, domotica, cyber security, cognitive computing, social networking, cloud analytics, mobile networking, privacy, open source, open data, open agent; - aumentare la trasversalità e la contaminazione tra i vari corsi di laurea, soprattutto nei corsi magistrali; - incrementare l'apprendimento delle soft skills attraverso la previsione di seminari con o senza riconoscimento di CFU; - rafforzare la formazione di base, soprattutto nelle lauree di primo livello; - aumentare la conoscenza dell'offerta formativa e rafforzare il collegamento con il mondo del lavoro prevedendo maggiori testimonianze aziendali in aula, stage, incontri con aziende, tesi di laurea in azienda, progetti di ricerca che coinvolgano anche studenti soprattutto della magistrale. Alla luce di quanto emerso si ritiene che i progetti formativi dei corsi della Facoltà siano adeguatamente strutturati al proprio interno. Si ritiene inoltre che le funzioni e le competenze che caratterizzano le figure professionali a cui preparano i vari corsi della Facoltà sono descritte in modo adeguato, e costituiscono quindi una base chiara per definire i risultati di apprendimento attesi e che i risultati di apprendimento attesi specifici e quelli generici previsti dall'ordinamento sono coerenti con le esigenze professionali, in modo che la preparazione dei laureati risponda ai più ampi bisogni della società e del mercato del lavoro (domanda di formazione). Le conclusioni dell'indagine per i corsi della Facoltà sono adeguate anche nello specifico per il Corso di Laurea in Informatica. In aggiunta si rileva, che la valutazione positiva da parte delle aziende dell'offerta formativa del corso di laurea è stata negli anni confermata dalla continua adesione agli incontri fra aziende e studenti che si tengono semestralmente, con presentazioni di opportunità di lavoro e di stage. Dal 2010 a oggi, e proseguendo una tradizione iniziata nel 1999, si sono svolti 16 incontri, a cui hanno partecipato in media quasi 20 aziende a incontro, per un totale di 132 aziende diverse, per una media di 2,32 partecipazioni ad azienda, con cinque aziende che hanno partecipato almeno 10 volte. Nell'ambito di questi incontri sono stati anche svolti mini-colloqui fra gli studenti e le aziende intervenute, per un totale di oltre 4000 colloqui. In corrispondenza di ognuno di questi eventi è stata svolta una tavola rotonda. Nel loro complesso, in questi anni esse hanno coperto vari temi relativi a tecnologie emergenti, tendenze del mercato del lavoro e

variazioni delle figure professionali nel mondo ICT.

## **Organizzazione e responsabilità della AQ del Cds**