

# PROGRAMMA DEL CORSO DI PROGRAMMAZIONE

## SETTORE SCIENTIFICO

ING-INF/05

## CFU

12

## DESCRIZIONE

MODULO INTRODUTTIVO: INFORMATICA DI BASE

- 1 - L'INFORMATICA COME SCIENZA DELLA RAPPRESENTAZIONE E DELL'INFORMAZIONE
- 2 - LA CODIFICA DELL'INFORMAZIONE
- 3 - L'ARCHITETTURA DI UN ELABORATORE
- 4 - L'HARDWARE
- 5 - I DISPOSITIVI DI MEMORIA
- 6 - IL SOFTWARE
- 7 - IL FILE SYSTEM
- 8 - LE RETI E INTERNET
- 9 - INTERNET: RETE DI RETI

ALGORITMI, LINGUAGGI E PROGRAMMI

- 1 - INTRODUZIONE ALLA PROGRAMMAZIONE
- 2 - ALGORITMI, LINGUAGGI E PROGRAMMI
- 3 - PROBLEM SOLVING
- 4 - FLOWCHART
- 5 - ESERCITAZIONE: PROBLEM SOLVING E FLOWCHART
- 6 - I LINGUAGGI DI PROGRAMMAZIONE

LINGUAGGIO C

- 1 - IL LINGUAGGIO C
- 2 - SVILUPPO, COMPILAZIONE ED ESECUZIONE
- 3 - IL PRIMO PROGRAMMA IN C
- 4 - PROGRAMMAZIONE INTERATTIVA
- 5 - ARITMETICA IN C
- 6 - ESERCIZI DI ARITMETICA IN C
- 7 - CONTROLLI CONDIZIONALI: IF
- 8 - PROGRAMMAZIONE STRUTTURATA
- 9 - ISTRUZIONE DI SELEZIONE IF...ELSE
- 10 - ISTRUZIONE DI ITERAZIONE WHILE
- 11 - ITERAZIONI CONTROLLATE DA CONTATORE
- 12 - ITERAZIONI CONTROLLATE DA SENTINELLA
- 13 - ITERAZIONI DI CONTROLLO ANNIDATE
- 14 - FLOAT E OPERATORI DI INCREMENTO
- 15 - ESERCITAZIONE CON ITERAZIONI
- 16 - ISTRUZIONI DI ITERAZIONE FOR
- 17 - ESERCITAZIONI CON CICLI FOR
- 18 - INPUT DI CARATTERI
- 19 - ISTRUZIONE DI SELEZIONE MULTIPLA SWITCH
- 20 - ISTRUZIONI DO...WHILE, BREAK E CONTINUE
- 21 - ESERCIZI DO, WHILE, E SWITCH
- 22 - OPERATORI LOGICI
- 23 - RIEPILOGO DELLA PROGRAMMAZIONE STRUTTURATA
- 24 - FUNZIONI DELLA LIBRERIA MATH
- 25 - FUNZIONI DEFINITE DAL PROGRAMMATORE
- 26 - PROTOTIPI E ATTRIBUTI DI FUNZIONE
- 27 - STACK E RECORD DI ATTIVAZIONE
- 28 - LIBRERIA STANDARD
- 29 - ENUM
- 30 - CLASSI DI MEMORIA

31 - ESERCITAZIONI CON CLASSI DI MEMORIA

32 - RICORSIONE

33 - ESEMPI DI RICORSIONE: FIBONACCI E HANOI

34 - ARRAY

35 - ESERCIZI CON ARRAY

36 - ARRAY E FUNZIONI

37 - ESERCIZI CON ARRAY E FUNZIONI

38 - PUNTATORI

39 - PUNTATORI E FUNZIONI

40 - ESERCITAZIONE CON PUNTATORI E SIZEOF

41 - ARITMETICA DEI PUNTATORI

42 - PUNTATORI E ARRAY

43 - PUNTATORI A FUNZIONI

44 - ESERCITAZIONE CON ARRAY DI PUNTATORI

45 - ESERCIZI CON PUNTATORI

PROGRAMMAZIONE OBJECT ORIENTED

1 - IL PARADIGMA OBJECT ORIENTED

2 - PROGETTAZIONE E OGGETTI SOFTWARE

3 - IL LINGUAGGIO DI MODELLAZIONE UML

4 - DIAGRAMMA DEI CASI D'USO

5 - PROPRIETÀ DEI CASI D'USO

6 - SOFTWARE OBJECT ORIENTED

7 - INTRODUZIONE ALLE CLASSI IN C++

8 - CLASSI STRING E VECTOR

9 - GESTIONE ECCEZIONI, DEI FILES E DELL'EREDITARIETÀ TRA CLASSI

10 - EREDITARIETÀ E POLIMORFISMO

11 - IL LINGUAGGIO JAVA

12 - PROGRAMMARE IN JAVA

13 - MATEMATICA E VERIFICHE SU SEQUENZE DI DATI

Il/La docente ha la facoltà di modificare in qualsiasi momento il programma didattico

### **OBIETTIVI FORMATIVI PER IL RAGGIUNGIMENTO DEI RISULTATI DI APPRENDIMENTO PREVISTI**

*\*\*/*

Obiettivo dell'insegnamento è fornire gli elementi di base di conoscenza delle tecniche di programmazione con particolare riferimento ad algoritmi e paradigmi di programmazione e dei linguaggi di programmazione.

### **MODALITÀ DI ESAME ED EVENTUALI VERIFICHE DI PROFITTO IN ITINERE**

Lo studente per superare l'esame può scegliere di fare l'esame orale presso la sede dell'Ateneo o la prova scritta in tutte le sedi di Italia, ivi compreso Roma. Per poter effettuare la prenotazione di esame lo studente deve aver seguito almeno l'80% delle video lezioni. Il test finale si compone di 31 domande a risposta multipla con 4 possibili risposte da redigere in 45 minuti.

Le domande di esame siano esse orali o scritte, coerentemente con i risultati di apprendimento attesi, sono finalizzate a misurare la preparazione acquisita in relazione a:

Conoscenza e capacità di comprensione attraverso domande specifiche relative alla gestione d'impresa ed alle scelte strategiche; Capacità di applicare conoscenza e comprensione con domande che consentano la valutazione rispetto a casi concreti; Autonomia di giudizio mediante domande che presuppongano la valutazione autonoma in ordine alla scelte da compiere ed alle soluzioni da proporre.

Gli esercizi e gli elaborati di Didattica Erogativa consentono invece di verificare i risultati di apprendimento raggiunti rispetto alle Abilità comunicative e alla Capacità di apprendimento.

Modalità di iscrizione e di gestione dei rapporti con gli studenti. L'iscrizione ed i rapporti con gli studenti sono gestiti mediante la piattaforma informatica che permette sia l'iscrizione ai corsi sia la partecipazione a forum e tutoraggi, il download del materiale didattico, la comunicazione con il docente. e' previsto un tutor che supporterà gli studenti durante il corso.

Attività di didattica erogativa (DE).

72 Videolezioni + 72 test di autovalutazione

Totale 72 ore.

Attività di didattica interattiva (DI).

Redazione di un elaborato su traccia del docente Partecipazione a una web conference Svolgimento delle prove in itinere con feedback Svolgimento della simulazione del test finale

Impegno totale stimato: 12 ore.

Attività di autoapprendimento.

Totale 216 ore

## RISULTATI DI APPRENDIMENTO ATTESI

Conoscenza e capacità di comprensione. Il corso intende fornire le conoscenze utili per comprendere metodi, tecniche e tecnologie per la progettazione e la implementazione di programmi. Esse includono: conoscenza e comprensione di fondamenti di problem solving, di algoritmi e strutture dati, di metodi e tecniche di astrazione; dei paradigmi di programmazione e dei linguaggi di programmazione, in particolare procedurali e ad oggetti; conoscenze dettagliate ed operative della programmazione ad oggetti.

Capacità di applicare conoscenza e comprensione. Il corso, trasferisce la capacità di progettare e implementare programmi, anche partendo dalla progettazione di alto livello del sistema software a cui appartengono; nonché la capacità di usare librerie e repository di programmi, piattaforme tecnologiche e framework di programmazione ; capacità di testing e debugging di programmi.

Autonomia di giudizio. Autonome capacità di giudizio sulla qualità dei programmi, sulla opportunità di riuso di programmi esistenti; nonché autonome capacità di giudizio, di valutazione comparativa e scelta di soluzioni, algoritmi e tools.

Abilità comunicative. Capacità di comunicare in team di programmazione e con i progettisti di sistemi software; con fornitori di tecnologie e servizi per ambienti di programmazione; capacità di trasferire all'utente conoscenze e procedure per l'uso di programmi. Lo sviluppo di abilità comunicative, sia orali che scritte, sarà anche stimolata attraverso la didattica interattiva (con la redazione di elaborati da parte dello studente) e i momenti di videoconferenza attivati, ivi compreso la prova finale di esame.

Capacità di apprendimento. Capacità di apprendere, sperimentare ed usare, in modo autonomo, linguaggi di programmazione anche di nuova definizione ed implementazione. Capacità di seguire l'evoluzione della programmazione in ogni suo aspetto attraverso la letteratura e la documentazione tecnica. Capacità di ricercare ed usare open e free software. Capacità di aggiornamento ed autoaggiornamento, anche attraverso courseware.