

PROGRAMMA DEL CORSO DI INFORMATICA

SETTORE SCIENTIFICO

ING-INF/05

CFU

12

FREQUENZA AL CORSO

Obbligatoria online

Ai corsisti viene richiesto di visionare almeno l'80% delle videolezioni presenti in piattaforma e di partecipare alla didattica interattiva.

VERIFICA DEGLI APPRENDIMENTI

La verifica degli apprendimenti avverrà attraverso un insieme di valutazioni intermedie ed un questionario (Prova scritta) finale. La verifica intermedia sarà incentrata sulla valutazione della capacità dell'allievo di contribuire attivamente al lavoro in piattaforma, attraverso interventi fondati, coerenti e originali nelle attività collaborative e interattive on-line. Saranno inoltre valutate abilità e competenze dell'allievo attraverso esercizi e project work, che saranno realizzati in piattaforma. Infine la verifica delle conoscenze teoriche acquisite dallo studente avverrà attraverso la somministrazione di un questionario durante un esame che si svolgerà in una sede Mercatorum alla presenza di una commissione d'esame. Il voto finale d'esame sarà calcolato attraverso il seguente schema:

Partecipazione al corso:

Frequenza, intensità e qualità delle interazioni tra il corsista e il docente e tra il corsista e i suoi colleghi in piattaforma.

Prova pratica:

Lo studente dovrà realizzare un progetto individuale o di gruppo (esercitazione e project work) che sarà valutato secondo una rubrica condivisa durante l'avvio delle lezioni.

Questionario finale:

Lo studente dovrà rispondere ad un questionario a risposta multipla di 30 domande con 4 opzioni ciascuna sugli argomenti trattati nel corso.

ARTICOLAZIONE DELLA DIDATTIVA INTERATTIVA

- 1 ora AREA FAQ
- 3 ore WEB-FORUM
- 3 ore ESERCIZI e PROJECT WORK
- 3 ore PROVE IN ITINERE CON FEEDBACK
- 2 ore SIMULAZIONE nel laboratorio virtualizzato di sistemi informatici

PROGRAMMA

• Storia dell'elaboratore e del calcolo automatico

- Dall'antichità al XIX secolo
- Dal '900 ai giorni nostri

• Codifica e rappresentazione dell'informazione

- I sistemi di numerazione posizionali
- Cambiamenti di base
- Il sistema di numerazione binario
- Operazioni algebriche

• Codifica dell'informazione multimediale

- Codifica delle immagini
- Codifica di video
- Codifica dei suoni
- Codifica dei testi
- Compressione

• Architettura di un elaboratore

Il modello di Von Neumann

- Il microprocessore
- I Bus
- La memoria centrale
- Le interfacce

• Le periferiche

- Hard disk
- Pen Drive
- Monitor
- Mouse
- Audio
- 8Video

• Algebra di Boole e circuiti logici

- Variabili booleane
- Tavole di verità
- Proprietà
- Circuiti logici

- **Sistemi operativi**

- Modello a strati di un sistema operativo
- Compiti di un sistema operativo
- Modello a 5 stati
- Il nucleo
- Il gestore della memoria
- Il gestore delle periferiche
- Il file system
- esempi

- **La sicurezza informatica**

- Classificazione dei pericoli
- Cifratura
- Esempi

- **La programmazione**

- Dal problema al programma
- Esecutore di algoritmi
- Confronto tra algoritmi
- Algoritmi numerici e non numerici
- La ricorsione
- Il linguaggio C: Struttura base di un programma
- Tipi di dati semplici
- L'ambiente IDE code::blocks
- Scrittura di semplici algoritmi numerici
- La ricorsione nel linguaggio C
- Tipi di dati strutturati
- Vettori e matrici - Gestione I/O (files)

- **Strutture dinamiche dei dati**

- Variabili puntatore
- Funzioni di gestione della memoria
- Liste concatenate
- Coda e pila

OBIETTIVI FORMATIVI

Il corso si propone di fornire conoscenze e competenze generali della computer science:

- Conoscere la storia del calcolo automatico
- Conoscere l'architettura di un computer
- Conoscere i sistemi di numerazione
- Conoscere il linguaggio binario
- Conoscere le tecniche principali di codifica di informazioni multimediali
- Conoscere le problematiche legate alla sicurezza informatica
- Saper individuare le componenti principali di un elaboratore elettronico
- Saper costruire test logici con variabili booleane (circuiti logici)
- Saper utilizzare il linguaggio binario ed i sistemi di numerazione
- Conoscere le basi della programmazione (programmazione strutturata)
- Saper costruire algoritmi numerici e non numerici semplici
- Conoscere la sintassi di un linguaggio di programmazione (C language)
- Saper trattare tipi di dati diversi
- Saper rappresentare in memoria tipi di dati strutturati
- Conoscere l'ambiente IDE Code::blocks- Saper effettuare il testing ed il debugging di un programma

LIBRI DI RIFERIMENTO

- **Dispense a cura del docente.**
- Dennis Curtin, Kim Foley, Kunal Sen, Cathy Morin. Informatica di base. Sesta Edizione. (2016). MacGraw-Hill.
- Marco Mezzalama, Elio Piccolo. Capire l'informatica. Dal microprocessore al Web 2.0. (2010). Ed. CittàStudi.
- Paolo Atzeni, Stefano Ceri, e al.. Basi di Dati. (2018). ED. MacGraw-Hill, Con Connect.
- **Paul J. Deitel, Harvey M. Deitel , e al.. Il linguaggio C. Fondamenti e tecniche di programmazione. Ediz. mylab. (2016). Con espansione online. ED. Pearson.**