



UNIVERSITÀ
degli STUDI
di CATANIA

DIPARTIMENTO DI INGEGNERIA ELETTRICA ELETTRONICA E
INFORMATICA

Corso di laurea in Ingegneria informatica

Anno accademico 2019/2020 - 1° anno

FONDAMENTI DI INFORMATICA A - Co

ING-INF/05 - 9 CFU - Insegnamento annuale

Docente titolare dell'insegnamento

VINCENZA CARCHIOLO

Email: vincenza.carchiolo@unict.it

Edificio / Indirizzo: Edificio 3 - piano V - stanza 11

Telefono: 095 7382359

Orario ricevimento: martedì e venerdì dalle ore 11 alle ore 12

OBIETTIVI FORMATIVI

corso introduce alla conoscenza dei principi dell'informatica e della programmazione di tipo procedurale.

Il corso ha l'obiettivo primario di fornire allo studente le conoscenze delle principali strutture dati, degli algoritmi di base e dei rudimenti di complessità computazionale.

Il corso inoltre fornisce allo studente le tecniche e gli strumenti per lo sviluppo di programmi applicativi mediante l'utilizzo del linguaggio di programmazione ANSI-C, con particolare attenzione ai meccanismi di "problem solving" e ricerca dell'errore

MODALITÀ DI SVOLGIMENTO DELL'INSEGNAMENTO

Il corso è organizzato in unità didattiche elementari in base ai contenuti e alle capacità che si intende sviluppare.

il corso prevede come metodo di insegnamento principale le lezioni frontali per acquisire le conoscenze teoriche di base e tutti gli elementi sintattici e lo svolgimento di esercitazioni proposte dal docente per acquisire la capacità di risolvere i problemi, applicare la conoscenza e utilizzare gli ambienti e le metodologie di sviluppo.

Il docente propone, inoltre, delle esercitazioni individuali che consistono nella soluzione di un problema che lo studente deve affrontare in autonomia che vengono successivamente corrette o discusse in classe.

PREREQUISITI RICHIESTI

Notazione esponenziale o scientifica. Arrotondamento dei valori numerici

Numeri reali. Operazioni con numeri reali. Potenza ad esponente razionale e reale. Logaritmo di un numero reale positivo

Concetto di funzione

Vettori e Matrici: Operazioni con vettori e matrici

FREQUENZA LEZIONI

Lo studente è tenuto a frequentare almeno il 70% delle lezioni del corso per poter sostenere le prove in itinere

La frequenza non è richiesta, seppure fortemente consigliata, per sostenere la prova di esame

CONTENUTI DEL CORSO

Modulo 1

- Elaborazione automatica dell'informazione e algoritmi - Algoritmi e programmi - Una notazione grafica per esprimere algoritmi - Linguaggi di programmazione - Il progetto di programmi
- Rappresentazione dell'informazione: Sistemi numerici - Conversione fra sistemi numerici - Sistema di numerazione binaria - Operazioni tra numeri binari - overflow e underflow - Rappresentazione dei numeri interi - Rappresentazione dei numeri con segno - Rappresentazione in virgola fissa e virgola mobile - Codici e Rappresentazione dei Caratteri - Algebra di Boole, Funzioni logiche, Espressione logiche, Applicazioni dell'algebra booleana
- Cenni della struttura di un elaboratore e sistema di elaborazione: Struttura di un elaboratore: memoria centrale, unità centrale, funzionamento elementare dell'elaboratore.
- Cenni sui sistemi di elaborazione: software di base: Traduzione ed esecuzione dei programmi - ambiente di programmazione - Linguaggi di programmazione: linguaggi imperativi - Compilatori, Linker, Cenni sulla realizzazione di applicazioni: Preprocessore - Commenti - Librerie e file di intestazione

Modulo 2

- Elementi fondamentali del linguaggio C: Sintassi del C - Struttura di un programma C - Compilazione di un programma - Tipi di dato e rappresentazioni - Tipi di dato principali - Identificatori - Variabili - Modificatori di accesso - Costanti - Operatori - Strutture di controllo - Istruzione di selezione - Istruzioni di Iterazione - Istruzioni di salto - Istruzioni di espressione - Istruzione blocco
- Array, stringhe e puntatori: Array monodimensionali - Puntatori ad array - Array come argomento di una funzione - Stringhe - Array di stringhe - Array multidimensionali - Variabili puntatore - Operatori ed espressioni con puntatori - Puntatori ad array - Puntatori a funzioni - Allocazione dinamica della memoria
- Funzioni: Regole di visibilità delle funzioni - Argomenti delle funzioni - Argomenti di main - Istruzione return - Valori restituiti da una funzione - Ricorsione - Dichiarazioni e campo di azione degli identificatori - Tecniche di legame dei parametri - effetti collaterali ed implementazione delle

funzioni

- Strutture, unioni e tipi definiti dall'utente: Strutture- Array di strutture - Strutture come argomenti di funzioni - Puntatori a strutture - Array e strutture all'interno di altre strutture - Unione - Enumerazioni - Sizeof - Typedef
- I/O da console da file: Lettura e scrittura di caratteri e stringhe da console - I/O formattato da console - Canali - File

Modulo 3

- Allocazione dinamica della memoria.
- Complessità Computazionale: Efficienza dei Programmi, le Notazioni O e W, Valutazione della Complessità di un Programma, Relazioni di Ricorrenza.
- Algoritmi di Ordinamento: Classi di algoritmi di ordinamento - ordinamento per selezione (selection sort) - algoritmi per inserzioni (insertion sort) -algoritmi per scambio bubble sort, quick sort, merge sort.
- Tipi di dato astratto: Liste, Code, Pile, Alberi binari, Alberi generali, Dizionari.

Il corso è organizzato in 7 Unità Didattiche Elementari (UDE) il cui contenuti è descritto negli argomenti del corso.

| Unità Didattica Elementare | Durata (ore) |
|-----------------------------------|---------------------|
| UDE 1 | 6 |
| UDE 2 | 12 |
| UDE 3 | 10 |
| UDE 4 | 10 |
| UDE 5 | 10 |
| UDE 6 | 8 |
| UDE 7 | 23 |

TESTI DI RIFERIMENTO

[Pel] Pellegrino Principe. C guida alla programmazione. Apogeo (isbn 9788850333288)

[BeGu] Bellini, Guidi. Linguaggio C. Guida alla programmazione. McGraw-Hill

ALTRO MATERIALE DIDATTICO

Il materiale didattico è pubblicato sul portale Studium di Ateneo

PROGRAMMAZIONE DEL CORSO

| Argomenti | Riferimenti testi |
|---|---|
| 1 UDE 1: Rappresentazione dei numeri interi e dei numeri reali. Algebra di boole, funzione ed espressioni logiche | Appunti del docente - [Pel] Capitolo 1, [Pel] Appendice C - [BeGu] Capitolo 1, 2 [BeGu] App. D, E |
| 2 UDE 1: Cenni sui sistemi di elaborazione; Traduzione ed esecuzione dei programmi; Ambiente di Programmazione; Linguaggi di Programmazione; Operazioni tra numeri binari; Codifica dei caratteri | Appunti del docente - [Pel] Capitolo 1, [Pel] Appendice C - [BeGu] Capitolo 1, 2 [BeGu] App. D, E |
| 3 UDE 2: Tipo di dato principale; Identificatori; Variabili; Modificatori di Accesso; Specificatori di classe di memorizzazione; Costanti; Operatori; Strutture di Controllo; Istruzioni di selezione, di iterazione, di salto, di espressione, di blocco | [Pel] Capitolo 2,4,5 - [BeGu] Capitolo 3, 4, 5, 6, 7, 8, 9 |
| 4 UDE 2: Algoritmi e programmi; Uso di notazione grafica per esprimere algoritmi; Espressioni Complesse | [Pel] Capitolo 2,4,5 - [BeGu] Capitolo 3, 4, 5, 6, 7, 8, 9 |
| 5 UDE 3: • Array Monodimensionali; Puntatori; Puntatori ad array; Stringhe; Strutture e strutture nidificate; Array di strutture | [Pel] Capitolo 3,7,8,11 - [BeGu] Capitolo 10, 13, 14, 16 |
| 6 UDE 3: Puntatori a strutture; Unione; Enumerazione | [Pel] Capitolo 3,7,8,11 - [BeGu] Capitolo 10, 13, 14, 16 |
| 7 UDE 4: Lettura e scrittura di caratteri e stringhe; I/O formattato; Canali; File di testo; Esercizi sugli argomenti svolti nelle UDE 1,2,3,4 | [Pel] Capitolo 11 - [BeGu] Capitolo 17, 19 [Pel] Capitolo 6,9,10 - [BeGu] Capitolo 11, 1 |
| 8 UDE 4: File Binari | [Pel] Capitolo 11 - [BeGu] Capitolo 17, 19 [Pel] Capitolo 6,9,10 - [BeGu] Capitolo 11, 15 |
| 9 UDE 5: Algoritmi di ricerca e ordinamento in memoria interna: classi di algoritmi; Esempi di Algoritmi di ordinamento | DISPENSE DEL DOCENTE |
| 10 UDE 5: Complessità computazionale; la notazione O e Omega grande; Cenni sulla valutazione della complessità di un programma | DISPENSE DEL DOCENTE |

| | | |
|----|--|----------------------|
| 11 | UDE 6: • Algoritmi di ricerca e ordinamento in memoria interna: classi di algoritmi • Esempi di Algoritmi di ordinamento | DISPENSE DEL DOCENTE |
| 12 | UDE 6: Complessità computazionale, la notazione O e omega, Cenni sulla valutazione della complessità di un programma | DISPENSE DEL DOCENTE |
| 13 | UDE 7: Strutture dati lineari: Liste, Pile, Code; Strutture dati annidate; Esercizi sugli argomenti svolti nelle UDE 5,6,7 | DISPENSE DEL DOCENTE |
| 14 | UDE 7: Strutture dati non lineari: Alberi, Hash | DISPENSE DEL DOCENTE |

VERIFICA DELL'APPRENDIMENTO

MODALITÀ DI VERIFICA DELL'APPRENDIMENTO

Modalità di verifica dell'apprendimento

Sono previste prove in itinere che permettono il superamento dell'esame.

L'esame finale (se necessario) è composto da una prova al calcolatore e da una prova orale opzionale.

Le competenze minime richieste per il superamento dell'esame sono le seguenti:

Conoscenze:

- Conoscenze elementari di paradigmi di programmazione
- Conoscenza della sequenza completa di produzione di programmi eseguibili
- Capacità di realizzazione di semplici conversioni decimali/binarie e vice versa
- Capacità di utilizzo di funzioni ed espressione booleane
- Comprensione delle differenti tipologie di algoritmi di ordinamento
- Capacità di saper applicare un qualunque algoritmo di ordinamento ad un problema informatico
- Capacità di utilizzare le strutture dati elementari (pile, code e liste)
- Capacità di risoluzione di problemi informatici tramite la decomposizione in funzioni elementari
- Sintassi e semantica del Linguaggio ANSI-C:

Capacità

- Capacità di risoluzione di problemi informatici elementari tramite l'utilizzo delle istruzioni di selezione ed iterazione
- Capacità di risoluzione di problemi informatici che richiedono l'utilizzo di vettori monodimensionali
- Capacità di utilizzo delle funzioni e dei parametri formali e dell'istruzione return

- Capacità di gestione della memoria Heap, tramite semplici operazioni di allocazione dinamica di elementi e vettori
- Capacità di utilizzo i file di testo per rendere persistenti le informazioni.
- Capacità di implementare semplici algoritmi di ordinamento utilizzando la memoria interna
- Capacità di implementare le strutture dati elementari (pile, code e liste)

I criteri adottati nella valutazione finale sono basati sulla pertinenza delle risposte rispetto alle domande formulate, sulla la qualità dei contenuti, sulla capacità di riportare esempi, e sulla proprietà di linguaggio tecnico e la capacità espressiva complessiva dello studente.

PROVE IN ITINERE

Sono previste due prove in itinere (durata 1:30-2:00 e 3:00-4:00 ore) durante il corso.

La prima prova in itinere è costituita da un elaborato al calcolatore che prevede la capacità di sviluppo di semplici programmi in C in accordo alle competenze erogate nelle Unità Didattiche 1,2,3,4) e da una sezione teorica relative alle Unità Didattiche 1,2,3

Il superamento della prima prova in itinere permette l'accesso alla seconda prova in itinere e, qualora superata, permette di acquisire fino a 8 (superamento della prova con voto pari a 4).

La seconda prova in itinere è un elaborato al calcolatore e da una sezione teorica (opzionale): essa permette di acquisire un voto massimo di 20 (superamento della prova con voto pari a 10). Il voto massimo ottenibile con le due prove in itinere è dunque di 28/30, dato dalla somma dei due punteggi.

Lo studente che abbia superato entrambe le prove in itinere ottenendo una valutazione complessiva di almeno 18/30, se lo desidera può integrare le due prove in itinere con la prova orale da svolgere negli appelli regolari (entro la fine dell'a.a.), al fine di poter ottenere una migliore valutazione. Lo studente che abbia superato entrambe le prove in itinere ottenendo una valutazione complessiva inferiore a 18/30, deve integrare le due prove in itinere con la prova orale da svolgere negli appelli regolari (entro la fine dell'a.a.), per il superamento dell'esame

PROVE DI FINE CORSO

La prova d'esame è composta da una prova scritta e una prova orale (opzionale) della durata di 3:00 - 4:00 ore.

La prova scritta è rappresentata da un elaborato al calcolatore (obbligatorio) e da una sezione teorica (opzionale) che prevede la conoscenza di tutti gli argomenti in programma. Il superamento della prova scritta permette di raggiungere un voto massimo di 27/30.

Lo studente che lo desidera può integrare la prova scritta con la prova orale da svolgere in un qualunque appello regolare (entro la fine dell'a.a.) per migliorare la propria valutazione, in questo caso non esistono soglie massime

ESEMPI DI DOMANDE E/O ESERCIZI FREQUENTI

Gli esercizi, esempi di prove in itinere o compiti d'esame sono disponibili su studium (studium.unict.it)
