

INSEGNAMENTO	Programmazione (M-Z)		
ANNO ACCADEMICO	2016/17		
CORSO DI LAUREA	Informatica e Tecnologie per la Produzione del Software		
DOCENTE	Prof.Giovanni Dimauro		
ANNO DI CORSO	primo	SEMESTRE	primo
CREDITI LEZIONI FRONTALI	9		
ORE LEZIONI FRONTALI	72		
ORE STUDIO INDIVIDUALE	153		
CREDITI ESERCITAZIONI	2		
ORE ESERCITAZIONI	30		
ORE STUDIO INDIVIDUALE	20		
CREDITI CASO DI STUDIO	1		
ORE LEZIONI FRONTALI	-		
ORE STUDIO INDIVIDUALE	25		
TOTALE CREDITI	12		

#### PREREQUISITI

Buona comprensione della Lingua Inglese. Lettura individuale da parte dello studente del 1<sup>^</sup> capitolo del testo adottato (v. sotto): INTRODUZIONE AI COMPUTER, A INTERNET E AL WEB

#### OBIETTIVI FORMATIVI

Il corso ha lo scopo di fornire i concetti base della programmazione attraverso i modelli e l'uso dei linguaggi imperativi. Al termine del corso lo studente dovrà essere in grado di analizzare e risolvere semplici problemi, progettando e sviluppando programmi strutturati nel linguaggio C.

OBIETTIVI PROFESSIONALIZZANTI //

## TESTO ADOTTATO

P. Deitel e H. Deitel, C - Corso completo di programmazione, Apogeo; 4 edizione (17 novembre 2010) - ISBN-10: 8850329547 - ISBN-13: 978-8850329540

a titolo di esempio si inviano 2 link (ciò non costituisce invito ad acquistare ai link indicati):

[https://books.google.it/books/about/C\\_Corso\\_completo\\_di\\_programmazione.html?id=Z-RfSQAACAAJ&redir\\_esc=y](https://books.google.it/books/about/C_Corso_completo_di_programmazione.html?id=Z-RfSQAACAAJ&redir_esc=y)

<http://www.isbns.co.ke/isbn/9788850329540>

## TESTO CONSIGLIATO

Kim N. King, Programmazione in C, Maggioli Editore, ISBN 8838785821

## PROPEDEUTICITÀ OBBLIGATORIE

Nessuna

## PROPEDEUTICITÀ CONSIGLIATE

Nessuna

## INCENTIVI ALLA FREQUENZA

L'eventuale lode potrà essere attribuita solo agli studenti che per la stragrande maggioranza delle lezioni hanno frequentato, interagito nel corso della lezione, proposto soluzioni e risolto i casi di studio proposti dal docente.

## ESONERI O PROVE DI VALUTAZIONE INTERMEDIE CON MODALITÀ DI ESECUZIONE

Una prova di esonero si tiene nella settimana di interruzione delle lezioni, normalmente collocata intorno alla metà di Novembre. La prova consiste nella soluzione di un problema individuando l'algoritmo e sviluppando il relativo programma in C.

## VALIDITÀ DEI VOTI E PREMIALITÀ ACQUISITE

Il voto ottenuto alla prova di esonero può essere speso esclusivamente nel primo appello utile, pertanto gli studenti sono fortemente invitati ad utilizzare il primo appello di Programmazione appositamente collocato immediatamente dopo la chiusura del corso. Ciò al fine di lasciare ampio margine per preparare e sostenere gli altri due esami del I semestre. Si confida e si stimola quindi lo studio costante anche nel corso delle vacanze natalizie con lo scopo di porre lo studente in condizione di sostenere i 3 esami del primo semestre prima dell'inizio del secondo.

## MODALITÀ DI ESECUZIONE DEGLI ESAMI

L'esame consiste nella soluzione di un problema individuando l'algoritmo e sviluppando il relativo programma in C (per chi non ha sostenuto/superato la prova di esonero) e nella prova orale (oppure scritta in forma di quiz a risposta chiusa/aperta) utile a verificare la preparazione teorica.

PROVA SCRITTA	No
PROVA ORALE	Si
PROVA DI LABORATORIO	Si
PROVA DI PROGETTO O CASO DI STUDIO	No

## CONTENUTI DEL CORSO

### CAPITOLO 2: INTRODUZIONE ALLA PROGRAMMAZIONE IN C

- 2.2 Un semplice programma C: visualizzare una riga di testo
- 2.3 Un altro semplice programma C: sommare due interi
- 2.4 Nozioni sulla memoria
- 2.5 L'aritmetica del C
- 2.6 Prendere delle decisioni: gli operatori di uguaglianza e relazionali

### CAPITOLO 3: LO SVILUPPO DI PROGRAMMI STRUTTURATI IN C

- 3.2 Gli algoritmi
- 3.3 Lo pseudocodice
- 3.4 Le strutture di controllo
- 3.5 Il comando di selezione if
- 3.6 Il comando di selezione if else
- 3.7 Il comando di iterazione while
- 3.8 Formulazione degli algoritmi: studio di un caso (l'iterazione controllata da un contatore)
- 3.9 Formulazione degli algoritmi con processo top-down per raffinamenti successivi: studio di un caso 2 (iterazione controllata da un valore sentinella)
- 3.10 Formulazione degli algoritmi con processo top-down per raffinamenti successivi: studio di un caso 3 (strutture di controllo nidificate)
- 3.11 Gli operatori di assegnamento
- 3.12 Gli operatori di incremento e di decremento

### CAPITOLO 4: IL CONTROLLO DEL PROGRAMMA IN C

- 4.2 Gli elementi dell'iterazione
- 4.3 Iterazione controllata da un contatore
- 4.4 Il comando di iterazione for
- 4.5 Il comando for: note e osservazioni
- 4.6 Esempi di utilizzo dei comando for
- 4.7 Il comando di selezione multipla switch

- 4.8 Il comando di iterazione do...while
- 4.9 Le istruzioni break e continue
- 4.10 Gli operatori logici
- 4.11 Operatori di uguaglianza (==) e di assegnamento (=)
- 4.12 Riassunto della programmazione strutturata

## CAPITOLO 5: LE FUNZIONI IN C

- 5.2 I moduli di programma in C
- 5.3 Le funzioni della libreria matematica
- 5.4 Le funzioni
- 5.5 Le definizioni di funzione
- 5.6 I prototipi di funzione
  
- 5.7 Lo stack delle chiamate di funzione e i record di attivazione
- 5.8 I file di intestazione
- 5.9 Invocare le funzioni: chiamata per valore e per riferimento
- 5.10 Generazione di numeri casuali
- 5.11 Esempio: un gioco d'azzardo
- 5.12 Le classi di memoria
- 5.13 Le regole di visibilità
- 5.14 La ricorsione
- 5.15 Esempio di utilizzo della ricorsione: la serie di Fibonacci
- 5.16 Ricorsione e iterazione

## CAPITOLO 6: I VETTORI IN C

- 6.2 I vettori
- 6.3 La dichiarazione dei vettori
- 6.4 Esempi sui vettori
- 6.5 Passare i vettori alle funzioni
- 6.6 L'ordinamento dei vettori
- 6.7 Studio di un caso: calcolare la media, mediana e la moda usando i vettori
- 6.8 La ricerca nei vettori
- 6.9 I vettori multidimensionali

## CAPITOLO 7: I PUNTATORI IN C

- 7.2 Dichiarazione e inizializzazione dei puntatori
- 7.3 Gli operatori sui puntatori
- 7.4 Passare gli argomenti per riferimento alle funzioni
- 7.5 Utilizzare il qualificatore const con i puntatori
- 7.6 L'ordinamento a bolle utilizzando una chiamata per riferimento
- 7.7 L'operatore sizeof
- 7.8 Le espressioni con i puntatori e l'aritmetica dei puntatori
- 7.9 La relazione tra i puntatori e i vettori
- 7.10 I vettori di puntatori
- 7.11 Studio di un caso: simulazione di un mescolatore e distributore di carte
- 7.12 I puntatori a funzioni

## CAPITOLO 11: L'ELABORAZIONE DEI FILE IN C

11.2 La gerarchia dei dati

11.3 1 file e gli stream

11.4 Creare un file ad accesso sequenziale

11.5 Leggere i dati da un file ad accesso sequenziale

11.6 1 file ad accesso casuale

11.7 Creare un file ad accesso casuale

11.8 Scrivere i dati in modo casuale in un file ad accesso casuale

11.9 Leggere i dati in modo casuale da un file ad accesso casuale

11.10 Studio di un caso: un programma per l'elaborazione delle transazioni