



UNIVERSITÀ DEGLI STUDI DI PALERMO

| | |
|--------------------------------|---|
| DEPARTMENT | Matematica e Informatica |
| ACADEMIC YEAR | 2015/2016 |
| BACHELOR'S DEGREE (BSC) | COMPUTER SCIENCE |
| INTEGRATED COURSE | PROGRAMMING AND LABORATORY - INTEGRATED COURSE |
| CODE | 05880 |
| MODULES | Yes |
| NUMBER OF MODULES | 2 |
| SCIENTIFIC SECTOR(S) | INF/01 |
| HEAD PROFESSOR(S) | SCIORTINO MARINELLA Professore Ordinario Univ. di PALERMO |
| OTHER PROFESSOR(S) | SCIORTINO MARINELLA Professore Ordinario Univ. di PALERMO LANGIU ALESSIO Professore a contratto Univ. di PALERMO |
| CREDITS | 12 |
| PROPAEDEUTICAL SUBJECTS | |
| MUTUALIZATION | |
| YEAR | 1 |
| TERM (SEMESTER) | 1° semester |
| ATTENDANCE | Not mandatory |
| EVALUATION | Out of 30 |
| TEACHER OFFICE HOURS | <p>LANGIU ALESSIO Thursday 11:30 12:39 da concordare all'interno del Dipartimento di Matematica e Informatica</p> <p>SCIORTINO MARINELLA Tuesday 13:00 14:00 Ufficio n. 201, Il piano del DMI Friday 12:00 13:30 Ufficio n. 201, Il piano del DMI</p> |

DOCENTE: Prof.ssa MARINELLA SCIORTINO

| | |
|---------------------------|---|
| PREREQUISITES | |
| LEARNING OUTCOMES | <p>Conoscenza e capacità di comprensione Acquisizione dei concetti fondamentali della programmazione strutturata, strutture dati elementari statiche e dinamiche, semplici algoritmi fondamentali di ordinamento o di ricerca, definizione ricorsiva di soluzioni. Padronanza dei costrutti fondamentali del linguaggio di programmazione C.</p> <p>Capacità di applicare conoscenza e comprensione Capacità di problem solving per semplici problemi numerici, di ricerca e ordinamento. Capacità di programmazione in linguaggio di programmazione C. Capacità di validare, mediante la scrittura di semplici programmi, i concetti appresi. Capacità di comprensione degli errori rilevati in fase di compilazione ed esecuzione di semplici programmi scritti in C. Capacità di decomporre problemi complessi in problemi più semplici da un punto di vista computazionale. Essere in grado di formulare strategie risolutive per semplici problemi con l'eventuale utilizzo di costrutti e strutture studiate durante il corso.</p> <p>Autonomia di giudizio Saper individuare le modalità più appropriate nel passaggio dei parametri. Saper confrontare due semplici programmi in termini di efficienza di calcolo e invarianza rispetto ai cambiamenti. Saper individuare una soluzione algoritmica efficiente di problemi semplici.</p> <p>Abilità comunicative Proprietà di espressione nella presentazione delle nozioni di base dei linguaggi di programmazione e della programmazione imperativa</p> <p>Capacità d'apprendimento Sapere approcciarsi ai vari linguaggi di programmazione e sapere contestualizzare le abilità acquisite in problemi concreti nell'ambito lavorativo.</p> |
| ASSESSMENT METHODS | Prova in itinere, Prova Scritta, Prova Pratica/Orale |
| TEACHING METHODS | Lezioni frontali in aula e in laboratorio |

**MODULE
STRUCTURED C PROGRAMMING**

Prof. ALESSIO LANGIU

SUGGESTED BIBLIOGRAPHY

A. Bellini, A. Guidi. Linguaggio C - guida alla programmazione. Mc Graw Hill.
H.M. Deitel, P.J. Deitel. Il Linguaggio C. Pearson.
Kim N. King. Programmazione in C. Apogeo
B. W. Kernighan, D. M. Ritchie. Il linguaggio C - Principi di Programmazione e Manuale di riferimento. Pearson Education Italia.

| | |
|-------------------------------|--------------------------------------|
| AMBIT | 50168-Formazione informatica di base |
| INDIVIDUAL STUDY (Hrs) | 102 |
| COURSE ACTIVITY (Hrs) | 48 |

EDUCATIONAL OBJECTIVES OF THE MODULE

Il modulo si propone di fornire allo studente gli strumenti teorici e pratici per la progettazione di un programma per il calcolatore elettronico nei suoi aspetti fondamentali: la rappresentazione dei dati e la formulazione di semplici algoritmi che fanno uso delle fondamentali strutture di controllo, di sequenza, selezione e iterazione. Il linguaggio di programmazione utilizzato è il C, per la sua diffusione e per essere di fatto paradigmatico rispetto alla maggior parte dei moderni linguaggi di programmazione.

SYLLABUS

| Hrs | Frontal teaching |
|------------|---|
| 2 | Introduzione al corso di Programmazione. Risoluzione dei problemi tramite un calcolatore. La nozione di Algoritmo. Esempi di algoritmi. Cenni sulla complessità di un algoritmo. |
| 6 | Il linguaggio C. Struttura di un programma in C. Identificatori. Programmi di input/output. Programmi che utilizzano il costrutto di sequenza. Le costanti e le variabili. Dichiarazione e assegnazione. Il tipo Int. La rappresentazione degli interi e degli interi relativi in binario. Il tipo char. Rappresentazione dei caratteri. Il codice ASCII e altri codici di caratteri. I tipi float e double. Rappresentazione dei numeri reali in memoria. Risoluzione di esercizi. |
| 12 | I costrutti di sequenza, selezione e iterazione in C. I costrutti di selezione. Il costrutto di selezione If...else. Il costrutto di selezione switch...case. Equivalenza dei cicli di iterazione. Gli operatori in C. Ordine di priorità degli operatori. I costrutti di iterazione: Il costrutto di iterazione for. Operatori di incremento e decremento di una variabile intera. Il costrutto di iterazione while, il costrutto while...do. Equivalenza dei costrutti di iterazione. Risoluzione di esercizi |
| 14 | Il tipo strutturato array. Array a una dimensione. Applicazioni. Codici per l'inserimento e la visualizzazione degli array. Array a più dimensioni. Matrici. Inserimento e visualizzazione di una matrice. Applicazioni degli array. Ricerca Lineare, ricerca binaria iterativa. Algoritmo di ordinamento Bubblesort. Le stringhe. Utilizzo della libreria <string.h>. Risoluzione di esercizi. |
| 4 | I file binari e testuali. Risoluzione di esercizi |
| 10 | Le funzioni in C. La dichiarazione, la definizione e la chiamata di funzioni. La visibilità. Il passaggio dei parametri. I puntatori. Array e puntatori. Aritmetica dei puntatori. Funzioni iterative su array. Risoluzione di esercizi. |

**MODULE
ABSTRACT DATA STRUCTURES**

Prof.ssa MARINELLA SCIORTINO

SUGGESTED BIBLIOGRAPHY

K. N. King. Programmazione in C. Apogeo.
P. Deitel, H. Deitel. Il linguaggio C. Fondamenti e tecniche di programmazione. Pearson.
A. Bellini, A. Guidi. Linguaggio C - guida alla programmazione. Mc Graw Hill.

| | |
|-------------------------------|--------------------------------------|
| AMBIT | 50168-Formazione informatica di base |
| INDIVIDUAL STUDY (Hrs) | 102 |
| COURSE ACTIVITY (Hrs) | 48 |

EDUCATIONAL OBJECTIVES OF THE MODULE

Il modulo si propone di fornire allo studente gli strumenti teorici e pratici per la progettazione di strategie ricorsive per la risoluzione di problemi. Si studierà la gestione dinamica della memoria e si approfondiranno semplici strutture dati dinamiche alcune delle quali definite mediante l'ausilio dei puntatori. Si implementeranno funzioni ricorsive ed iterative per la gestione di tali strutture.

SYLLABUS

| Hrs | Frontal teaching |
|------------|---|
| 8 | La ricorsione. Esempi di funzioni ricorsive: il fattoriale, la somma di una successione di interi, i numeri di Fibonacci. Confronto tra iterazione e ricorsione. Esempi di funzioni ricorsive su array: ricerca binaria ricorsiva e algoritmo di ordinamento Quicksort. |
| 8 | Esempi di strategie ricorsive per la soluzione di problemi su array e su stringhe |
| 8 | Puntatori e oggetti dinamici. Allocazione e de-allocazione di memoria. Le strutture. Strutture e puntatori. Tipi derivati composti tramite struttura. |
| 4 | Esempi di strategie risolutive per la gestione di strutture dinamiche o statiche anche tramite l'uso di file testuali o binari. |
| 10 | Strutture Dati Astratte (ADS). Una semplice ADS: la lista. Implementazione tramite array. Implementazione tramite lista concatenata. Operazioni (iterative e ricorsive) di inserimento, ricerca e cancellazione di elementi in liste concatenate. Cenni ad altre strutture dati astratte: la pila e la coda. |
| 10 | Esempi di problemi modellabili attraverso l'uso di semplici ADS e strategie iterative e ricorsive per la risoluzione di tali problemi. |