



UNIVERSITÀ DEGLI STUDI DI PALERMO

DEPARTMENT	Ingegneria
ACADEMIC YEAR	2015/2016
BACHELOR'S DEGREE (BSC)	MANAGEMENT AND COMPUTER ENGINEERING
SUBJECT	FUNDAMENTALS OF PROGRAMMING
TYPE OF EDUCATIONAL ACTIVITY	A
AMBIT	50283-Matematica, informatica e statistica
CODE	01727
SCIENTIFIC SECTOR(S)	ING-INF/05
HEAD PROFESSOR(S)	PERI DANIELE Ricercatore Univ. di PALERMO
OTHER PROFESSOR(S)	
CREDITS	12
INDIVIDUAL STUDY (Hrs)	192
COURSE ACTIVITY (Hrs)	108
PROPAEDEUTICAL SUBJECTS	
MUTUALIZATION	
YEAR	1
TERM (SEMESTER)	1° semester
ATTENDANCE	Not mandatory
EVALUATION	Out of 30
TEACHER OFFICE HOURS	PERI DANIELE Wednesday 15:00 - 16:00 Ricevimento in modalita a distanza sulla piattaforma MS Teams

DOCENTE: Prof. DANIELE PERI

PREREQUISITES	
LEARNING OUTCOMES	<p>Conoscenza e capacità di comprensione Lo studente acquisirà approfondita conoscenza della programmazione strutturata in linguaggio C. Conoscerà i principali strumenti di programmazione in ambiente Unix-like. Acquisirà elementi di rappresentazione delle informazioni nei calcolatori e metodologie di base per la progettazione e l'analisi di reti logiche combinatorie e sequenziali. Lo studente acquisirà conoscenze di base sulle architetture dei calcolatori e sui sistemi operativi Unix-like.</p> <p>Capacità di applicare conoscenza e comprensione Lo studente sarà in grado di valutare le possibili soluzioni software a problemi di complessità media e affrontarne l'implementazione utilizzando strumenti e ambienti di sviluppo per la programmazione in linguaggio C in ambienti Unix-like. Sarà in grado di affrontare semplici problemi di rappresentazione binaria delle informazioni. Sarà in grado di progettare a livello funzionale circuiti logici per la soluzione di semplici problemi.</p> <p>Autonomia di giudizio Lo studente sarà in grado di affrontare in autonomia l'analisi, la progettazione e l'implementazione di software utilizzando la programmazione strutturata. Sarà in grado di valutare la qualità del software in termini di semplicità, leggibilità, strutturazione, efficienza e riutilizzabilità.</p> <p>Abilità comunicative Lo studente sarà in grado di esporre, efficacemente e con proprietà di linguaggio, analisi e soluzioni di problemi affrontabili con la programmazione strutturata e con la progettazione funzionale di circuiti logici, nonché di problemi di rappresentazione delle informazioni.</p> <p>Capacità d'apprendimento Lo studente sarà in grado di affrontare in maniera autonoma problemi di programmazione strutturata individuando e integrando soluzioni parziali già disponibili, sia formalizzate sia implementate. Sarà in grado di approfondire in autonomia la conoscenza di moduli software e interfacce di programmazione. Sarà in grado di approfondire la conoscenza dei linguaggi e paradigmi di programmazione, dei sistemi operativi, delle architetture dei calcolatori e dei circuiti logici.</p>
ASSESSMENT METHODS	Prove scritte (in itinere, finale), prova orale
EDUCATIONAL OBJECTIVES	Al termine del corso lo studente conoscerà i concetti di base necessari alla comprensione della struttura dei calcolatori elettronici digitali programmabili. Conoscerà le principali nozioni sull'algebra di Boole e sulle reti logiche. Avrà conoscenza delle problematiche inerenti le metodologie di progettazione di reti logiche combinatorie e sequenziali. Avrà conoscenza di base di sistemi operativi Unix-like. Lo studente sarà in grado di valutare, analizzare, comunicare e implementare le possibili soluzioni software a problemi applicativi di media complessità utilizzando l'acquisita padronanza del linguaggio C.
TEACHING METHODS	Lezioni frontali Esercitazioni in aula e nelle aule informatiche
SUGGESTED BIBLIOGRAPHY	J. Glenn Brookshear, Stephen G. Kochan, "Fondamenti di informatica e programmazione in C", Pearson B. W. Kernighan, D. M. Ritchie, "Linguaggio C", Pearson M. Morris Mano, Charles R. Kime, "Reti Logiche", Pearson

SYLLABUS

Hrs	Frontal teaching
2	Calcolatori elettronici digitali programmabili e loro evoluzione. Struttura di un calcolatore. Modello Von Neumann: CPU, memoria, input/output. Microprocessori.
2	Rappresentazione delle informazioni. Rappresentazione interna ed esterna. Sistemi di numerazione. Numerazione binaria. Bit, byte e multipli. Conversioni binario-decimale. Operazioni aritmetiche. Sistema di numerazione ottale. Sistema di numerazione esadecimale. Codici. Codice BCD. Rappresentazione di interi con segno.
2	Rappresentazione delle informazioni. Rappresentazione in complemento alla base. Rappresentazione di numeri reali: virgola fissa, virgola mobile. Rappresentazione di caratteri alfanumerici. Codice ASCII. Rappresentazione di immagini (cenni).
2	Algebra Booleana. Operatori e porte logiche. Funzioni. Tabelle di verità. Diagrammi e circuiti logici. Identità fondamentali. Principio di dualità. Teorema di de Morgan. Complemento di una funzione. Forme canoniche. Mintermini. Maxtermini. Sintesi a due livelli.
2	Algebra Booleana. Mappe di Karnaugh. Implicanti, primi implicanti e primi implicanti essenziali di funzioni booleane. Minimizzazione di funzioni booleane. Operatore XOR. Operatori funzionalmente completi.
2	Reti combinatorie. Decoder ed encoder. Espansione in serie di decoder. Encoder con priorità. Multiplexer e demultiplexer.

SYLLABUS

Hrs	Frontal teaching
2	Reti combinatorie. Sintesi con decoder. Sintesi con multiplexer. Sommatore.
2	Reti sequenziali. Modelli di Mealy e Moore. Latch. Flip-Flop.
2	Reti sequenziali. Sintesi di reti sequenziali sincrone.
2	Software di base e software applicativo. Sistemi operativi. Astrazione dall'hardware. Driver. File system. Processi. Sistemi Unix-like. Terminale, shell e riga di comando.
2	Linguaggi di programmazione e sviluppo del software. Linguaggio macchina. Linguaggi compilati e interpretati. Codice sorgente, codice oggetto e codice eseguibile. Fasi della compilazione. Preprocessore, compilatore, assembler e linker. Moduli e librerie. Caricamento in memoria ed esecuzione dei programmi. Argomenti della riga di comando e codice di uscita.
2	Programmazione strutturata. Variabili e tipi di dati. Strutture dati. Costanti e literal. Istruzioni di assegnamento. Istruzioni di controllo. Unità procedurali: procedure, parametri e funzioni.
2	Astrazioni sui dati. Strutture dati elementari. Array. Liste, stack e code. Alberi. Strutture statiche e dinamiche. Tipi di dati definiti dall'utente. Tipi di dati astratti.
2	Programmazione strutturata e linguaggio C. Struttura dei programmi in linguaggio C. Direttive di preprocessore. Istruzioni e blocchi. Funzioni. Variabili e costanti. Tipi di dati predefiniti. Funzione principale. Input e output. Input/output standard. Input e output formattato. Libreria standard.
2	Programmazione strutturata e linguaggio C. Espressioni. Operatori aritmetici. Operatori relazionali e logici. Operatori bit a bit. Conversioni di tipo. Espressioni condizionali. Precedenza e ordine di valutazione.
2	Programmazione strutturata e linguaggio C. Operatori di incremento e decremento. Cicli iterativi: for, while e do-while. Istruzioni per l'interruzione e la continuazione dei cicli.
2	Programmazione strutturata e linguaggio C. Strutture di controllo. Costrutto if-else. Costrutto switch.
2	Programmazione strutturata e linguaggio C. Definizioni e prototipi di funzioni. Parametri formali e variabili automatiche. Stack di esecuzione. Passaggio dei parametri per copia e per riferimento. Ricorsione.
2	Programmazione strutturata e linguaggio C. Puntatori, indirizzi e gestione della memoria. Aritmetica degli indirizzi. Puntatori a puntatori. Puntatori generici.
2	Programmazione strutturata e linguaggio C. Stringhe. Array. Allocazione dinamica di memoria. Spostamento di dati in memoria. Duplicazione di stringhe.
2	Programmazione strutturata e linguaggio C. Tipi aggregati. Strutture. Campi di bit. Union. Definizione di sinonimi per i tipi (typedef).
2	Programmazione strutturata e linguaggio C. File. Principali operazioni di accesso ai file.
2	Programmazione strutturata e linguaggio C. Preprocessore. File header. Inclusione. Compilazione condizionale.
2	Programmazione strutturata e linguaggio C. La libreria standard. Funzioni matematiche.
2	Programmazione strutturata e linguaggio C. Puntatori a funzioni.
2	Algoritmi e strutture dati. Ricerca di valori in un array: ricerca lineare e ricerca dicotomica. Algoritmi di ordinamento basati sui confronti: ordinamento per selezione, ordinamento per inserimento, mergesort. Algoritmi di ordinamento non basati sui confronti: Integersort.
2	Algoritmi e strutture dati. Stack e espressioni in notazione polacca inversa (RPN).
2	Algoritmi e strutture dati. Operazioni di base su liste concatenate: creazione di un elemento, aggiunta in testa, scansione della lista.
2	Algoritmi e strutture dati. Operazioni di base su liste: concatenazione, aggiunta in coda, cancellazione. Operazioni avanzate su liste: inversione, ordinamento, map.
2	Algoritmi e strutture dati. Alberi binari. Alberi binari di ricerca. Alberi n-ari. Ricerca in profondità.
2	Algoritmi e strutture dati. Grafi. Visite di grafi

Hrs	Practice
2	Rappresentazione delle informazioni. Algebra booleana.
2	Reti combinatorie.
2	Reti sequenziali.
2	CPU e linguaggio macchina.
2	Programmazione strutturata e linguaggio C. Struttura dei programmi in linguaggio C. Direttive di preprocessore. Istruzioni e blocchi. Funzioni. Variabili e costanti. Tipi di dati predefiniti. Funzione principale. Input e output. Input/output standard. Input e output formattato. Libreria standard.
2	Programmazione strutturata e linguaggio C. Espressioni. Operatori aritmetici. Operatori relazionali e logici. Operatori bit a bit. Conversioni di tipo. Espressioni condizionali. Precedenza e ordine di valutazione.
2	Programmazione strutturata e linguaggio C. Operatori di incremento e decremento. Cicli iterativi: for, while e do-while. Istruzioni per l'interruzione e la continuazione dei cicli.
2	Programmazione strutturata e linguaggio C. Strutture di controllo. Costrutto if-else. Costrutto switch.
2	Programmazione strutturata e linguaggio C. Definizioni e prototipi di funzioni. Parametri formali e variabili automatiche. Stack di esecuzione. Passaggio dei parametri per copia e per riferimento. Ricorsione.

Hrs	Practice
2	Programmazione strutturata e linguaggio C. Puntatori, indirizzi e gestione della memoria. Aritmetica degli indirizzi. Puntatori a puntatori. Puntatori generici.
2	Programmazione strutturata e linguaggio C. Stringhe. Array. Allocazione dinamica di memoria. Spostamento di dati in memoria. Duplicazione di stringhe.
2	Programmazione strutturata e linguaggio C. Tipi aggregati. Strutture. Campi di bit. Union. Definizione di sinonimi per i tipi (typedef).
2	Programmazione strutturata e linguaggio C. File. Principali operazioni di accesso ai file.
2	Programmazione strutturata e linguaggio C. Preprocessore. File header. Inclusione. Compilazione condizionale.
2	Programmazione strutturata e linguaggio C. La libreria standard. Funzioni matematiche.
2	Programmazione strutturata e linguaggio C. Puntatori a funzioni.
2	Algoritmi e strutture dati. Ricerca di valori in un array: ricerca lineare e ricerca dicotomica. Algoritmi di ordinamento basati sui confronti: ordinamento per selezione, ordinamento per inserimento, mergesort. Integersort.
2	Algoritmi e strutture dati. Stack. Stack ed espressioni in notazione polacca inversa (RPN).
2	Algoritmi e strutture dati. Operazioni di base su liste concatenate: creazione di un elemento, aggiunta in testa, scansione della lista.
2	Algoritmi e strutture dati. Operazioni di base su liste: concatenazione, aggiunta in coda, cancellazione. Operazioni avanzate su liste: inversione, ordinamento, map.
2	Algoritmi e strutture dati. Alberi binari. Alberi binari di ricerca. Alberi n-ari. Ricerca in profondità.
2	Algoritmi e strutture dati. Grafi. Visite di grafi.
2	Scrittura di codice modulare. Compilazione di programmi composti da più file. Compilazione condizionale. Make e makefile. Librerie.