

FACOLTÀ	Ingegneria
ANNO ACCADEMICO	2012/2013
CORSO DI LAUREA	Ingegneria Informatica e delle Telecomunicazioni
INSEGNAMENTO	Calcolatori Elettronici
TIPO DI ATTIVITÀ	Di base
AMBITO DISCIPLINARE	Matematica, informatica e statistica
CODICE INSEGNAMENTO	01727
ARTICOLAZIONE IN MODULI	NO
NUMERO MODULI	
SETTORI SCIENTIFICO DISCIPLINARI	ING-INF/05
DOCENTE RESPONSABILE	Valeria Seidita Ricercatore Università degli Studi di Palermo
CFU	12
NUMERO DI ORE RISERVATE ALLO STUDIO PERSONALE	190
NUMERO DI ORE RISERVATE ALLE ATTIVITÀ DIDATTICHE ASSISTITE	110
PROPEDEUTICITÀ	Nessuna
ANNO DI CORSO	Primo
SEDE DI SVOLGIMENTO DELLE LEZIONI	Consultare il sito www.ingegneria.unipa.it
ORGANIZZAZIONE DELLA DIDATTICA	Lezioni frontali Esercitazioni in aula o nelle aule informatiche
MODALITÀ DI FREQUENZA	Facoltativa
METODI DI VALUTAZIONE	Prova Scritta e Orale
TIPO DI VALUTAZIONE	Voto in trentesimi
PERIODO DELLE LEZIONI	Primo e secondo semestre
CALENDARIO DELLE ATTIVITÀ DIDATTICHE	Consultare il sito www.ingegneria.unipa.it
ORARIO DI RICEVIMENTO DEGLI STUDENTI	Martedì e Venerdì dalle ore 9:00 alle or 10:00 e per appuntamento

RISULTATI DI APPRENDIMENTO ATTESI

Conoscenza e capacità di comprensione

Lo studente alla fine del corso acquisirà una buona conoscenza della programmazione in linguaggio C. Sarà in grado di valutare, analizzare e sintetizzare le possibili soluzioni software a semplici problemi. Avrà conoscenza delle problematiche inerenti le metodologie di progettazione di reti logiche combinatorie e le metodologie di progettazione di reti sequenziali. Inoltre avrà una conoscenza di base dell'architettura del calcolatore.

Capacità di applicare conoscenza e comprensione

Lo studente sarà in grado di utilizzare strumenti e ambienti di sviluppo per la programmazione C e di implementare semplici programmi. Sarà in grado di progettare a livello logico funzionale semplici circuiti logici per la soluzione di problemi elementari. Sarà in grado di progettare semplici strumenti software.

Autonomia di giudizio

Lo studente sarà in grado sia di effettuare l'analisi di un problema che di progettare, a partire da una descrizione verbale, una opportuna soluzione software. Sarà in grado di valutarne la qualità di una soluzione software in termini di semplicità, leggibilità, efficienza e possibilità di riutilizzo. Sarà in grado di capire i principi di funzionamento del calcolatore.

Abilità comunicative

Lo studente acquisirà la capacità di comunicare ed esprimere problematiche inerenti l'oggetto del corso. Sarà in grado di sostenere conversazioni su tematiche relative alla realizzazione di circuiti logici. Sarà in grado di utilizzare un linguaggio semplice e chiaro per la descrizione dei processi di analisi e di sintesi di soluzioni software a problemi elementari.

Capacità d'apprendimento

Lo studente dovrà sviluppare la capacità di apprendere i processi di analisi e di sintesi relativi alla codifica di programmi di complessità medio-bassa e relativi alla progettazione di semplici circuiti logici.

OBIETTIVI FORMATIVI

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. 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.

Numero di ore programmato in funzione degli argomenti del corso:

	LEZIONI FRONTALI
4	Calcolatori elettronici digitali programmabili e loro evoluzione. Struttura di un calcolatore. Modello Von Neumann: CPU, memoria, input/output. Microprocessori.
6	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. Rappresentazione in complemento alla base. Rappresentazione di numeri reali: virgola fissa, virgola mobile. Rappresentazione di caratteri alfanumerici. Codice ASCII. Rappresentazione di immagini (cenni).
6	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. Mappe di Karnaugh. Implicanti, primi implicanti e primi implicanti essenziali di funzioni booleane. Minimizzazione di funzioni booleane. Xor. Operatori funzionalmente completi.
4	Reti combinatorie. Decoder ed encoder. Espansione in serie di decoder. Encoder con priorità. Multiplexer e demultiplexer. Sintesi con decoder. Sintesi con multiplexer. Sommatore.
4	Reti sequenziali. Modelli di Mealy e Moore. Latch. Flip-Flop. Sintesi di reti sequenziali sincrone.
6	Software di base. Sistemi operativi. Astrazione dall'hardware. Driver. File system. Introduzione ai sistemi operativi.
4	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 di un programma. Argomenti della riga di comando e codice di uscita.
20	<p>Programmazione strutturata e linguaggio C</p> <p>Struttura dei programmi in linguaggio C. Direttive di preprocessore. Istruzioni blocchi. Funzioni. Variabili e costanti. Tipi di dati predefiniti. Funzione principale. Espressioni. Operatori aritmetici. Operatori relazionali e logici. Conversioni di tipi. Operatori di incremento e decremento. Operatori bit a bit. Espressioni condizionali. Precedenza e ordine di valutazione.</p> <p>Strutture di controllo. Costrutto if-else. Costrutto switch. Cicli for, while e do-while. Istruzioni per l'interruzione e la continuazione dei cicli.</p> <p>Definizioni e prototipi di funzioni. Parametri formali e variabili automatiche. Scelta dell'ordine di esecuzione. Passaggio dei parametri per copia e per riferimento. Ricorsione.</p> <p>Puntatori, indirizzi e gestione della memoria. Aritmetica degli indirizzi. Puntatori a puntatori. Puntatori generici. Puntatori a funzioni. Stringhe. Array. Allocazione dinamica di memoria. Spostamento di dati in memoria. Duplicazione di stringhe.</p> <p>Tipi aggregati. Strutture. Campi di bit. Union. Definizione di sinonimi per i tipi (typedef).</p> <p>Input e output. L'input/output standard. Input e output formattato. File. Principali operazioni di accesso ai file.</p> <p>Preprocessore. Header file. Inclusione. Compilazione condizionale.</p> <p>La libreria standard. Funzioni matematiche.</p> <p>Modularizzazione del codice. Compilazione di programmi composti da più file. Make e Makefile (cenni).</p>
20	<p>Algoritmi e strutture dati astratte</p> <p>Ricerca di valori in un array: ricerca lineare e ricerca dicotomica.</p> <p>Algoritmi di ordinamento basati sui confronti: ordinamento per selezione, ordinamento per inserimento, mergesort. Integersort.</p> <p>Stack. Liste. Alberi binari. Alberi binari di ricerca. Alberi n-ari. Ricerca in profondità. Grafi. Visita di un grafo.</p>
	ESERCITAZIONI
2	Rappresentazione delle informazioni.
2	Algebra booleana
4	Reti combinatorie
4	Reti sequenziali
2	Linguaggi di programmazione e sviluppo del software
22	Programmazione strutturata e linguaggio C. Algoritmi e strutture dati.
TESTI CONSIGLIATI	<ul style="list-style-type: none"> – J. Glenn Brookshear. Informatica. Una panoramica generale. Pearson – Stephen G. Kochan. PROGRAMMARE IN C. Pearson – M. Morris Mano, Charles R. Kime. Reti logiche. Pearson – Dennis M. Ritchie, Brian W. Kernighan. Il linguaggio C. Pearson