

<b>FACOLTÀ</b>	Ingegneria
<b>ANNO ACCADEMICO</b>	2013/2014
<b>CORSO DI LAUREA (o LAUREA MAGISTRALE)</b>	Ingegneria Informatica e delle Telecomunicazioni Classe L-8 – Lauree in Ingegneria dell'informazione
<b>INSEGNAMENTO</b>	Calcolatori Elettronici
<b>TIPO DI ATTIVITÀ</b>	Di base
<b>AMBITO DISCIPLINARE</b>	Matematica, informatica e statistica
<b>CODICE INSEGNAMENTO</b>	01727
<b>ARTICOLAZIONE IN MODULI</b>	SI
<b>NUMERO MODULI</b>	4
<b>SETTORI SCIENTIFICO DISCIPLINARI</b>	ING-INF/05
<b>DOCENTE RESPONSABILE</b>	Valeria Seidita Ricercatore Università degli Studi di Palermo <a href="mailto:valeria.seidita@unipa.it">valeria.seidita@unipa.it</a>
<b>CFU</b>	12
<b>NUMERO DI ORE RISERVATE ALLO STUDIO PERSONALE</b>	192
<b>NUMERO DI ORE RISERVATE ALLE ATTIVITÀ DIDATTICHE ASSISTITE</b>	108
<b>PROPEDEUTICITÀ</b>	Conoscenza di matematica di base.
<b>ANNO DI CORSO</b>	Primo
<b>SEDE DI SVOLGIMENTO DELLE LEZIONI</b>	Consultare il sito <a href="http://www.unipa.it">www.unipa.it</a>
<b>ORGANIZZAZIONE DELLA DIDATTICA</b>	Lezioni frontali; Esercitazioni in aula o nelle aule informatiche; Esercitazioni di gruppo per progetti e implementazioni; Presentazioni e discussioni in aula di progetti e implementazioni; Dibattiti guidati in aula sugli argomenti del corso.
<b>MODALITÀ DI FREQUENZA</b>	Facoltativa
<b>METODI DI VALUTAZIONE</b>	Prova Scritta e Orale
<b>TIPO DI VALUTAZIONE</b>	Voto in trentesimi
<b>PERIODO DELLE LEZIONI</b>	Primo e secondo semestre
<b>CALENDARIO DELLE ATTIVITÀ DIDATTICHE</b>	Consultare il sito <a href="http://www.unipa.it">www.unipa.it</a>
<b>ORARIO DI RICEVIMENTO DEGLI STUDENTI</b>	Martedì e Venerdì dalle ore 9:00 alle ore 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.

Per il raggiungimento di questo obiettivo il corso comprende lezioni frontali, analisi e discussione di pattern di problemi risolti in forma algoritmica e con l'uso del linguaggio C, di casi di studio noti nell'ambito delle reti logiche quali ad esempio multiplexer, decodificatori, etc. e infine dibattiti e studio guidato individuale, o a gruppi, su temi di approfondimento della materia.

La verifica di questo obiettivo prevede un esame scritto ed uno orale comprendenti la discussione degli argomenti, del programma e la presentazione degli argomenti approfonditi.

### **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.

Per il raggiungimento di questo obiettivo il corso comprende esercitazioni in aula sia individuali che di gruppo e la preparazione (sia individualmente che in gruppo) autonoma di un argomento assegnato dal docente su tematiche inerenti gli argomenti del corso.

Per la verifica di questo obiettivo una parte dell'esame scritto sarà organizzata in modo da sottoporre allo studente quesiti riguardanti la programmazione in C e la progettazione di reti logiche. Inoltre lo studente verrà valutato durante le esercitazioni sul grado di comprensione acquisita nel preparare l'argomento assegnato per un massimo di 4/30 sulla valutazione totale.

### **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 in termini di semplicità, leggibilità, efficienza e possibilità di riutilizzo. Sarà in grado di capire i principi di funzionamento del calcolatore.

Per il raggiungimento di questo obiettivo il corso prevede l'analisi e la discussione in aula di casi di studio legati alla soluzione con algoritmi particolari di problemi comuni; lezioni ed esercitazioni di gruppo sulla progettazione di circuiti logici e sulla implementazione di algoritmi.

Per la verifica di questo obiettivo l'esame comprende la discussione sui casi di studio e sui progetti preparati e svolti durante le esercitazioni in aula.

### **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.

Per il raggiungimento di questo obiettivo il corso si articola in esercitazioni di gruppo e discussioni sulla progettazione ed implementazione di algoritmi e reti logiche e la preparazione autonoma (sia individualmente che in gruppo) di un argomento assegnato dal docente su tematiche inerenti gli argomenti del corso.

Per la verifica di questo obiettivo verrà valutata durante l'esame scritto e l'esame orale la capacità dello studente di esporre in maniera chiara ed esaustiva i concetti legati agli argomenti del corso ed inoltre durante le esercitazioni in aula verranno valutati (per un massimo di 4/30 sulla valutazione totale) il grado di autonomia e chiarezza maturati nell'esporre gli argomenti assegnati.

### **Capacità d'apprendimento**

Lo studente svilupperà 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.

Per il raggiungimento di questo obiettivo il corso si articola in esercitazioni di gruppo e discussioni sulla progettazione ed implementazione di algoritmi e reti logiche e la preparazione (sia individualmente che in gruppo) autonoma di un argomento assegnato dal docente su tematiche inerenti gli argomenti del corso.

La verifica di questo obiettivo prevede un esame scritto ed uno orale comprendenti la discussione degli argomenti del programma e la presentazione degli argomenti approfonditi.

### **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:

	<b>LEZIONI FRONTALI</b>
2	Struttura di un calcolatore. Modello Von Neumann: CPU, memoria, input/output..
4	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.
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. Mappe di Karnaugh. Minimizzazione di funzioni booleane. Xor. Operatori funzionalmente completi.
3	Reti combinatorie. Decoder ed encoder. Encoder con priorità. Multiplexer e demultiplexer. Sintesi con decoder. Sintesi con multiplexer. Sommatore.
2	Reti sequenziali. Modelli di Mealy e Moore. Latch. Flip-Flop. Sintesi di reti sequenziali sincrone.
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.
18	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. Espressioni. Operatori aritmetici. Operatori relazionali e logici. Conversioni di tipo. Operatori di incremento e decremento. Operatori bit a bit. Espressioni condizionali. Precedenza e ordine di valutazione.

	<p>Strutture di controllo.  Definizioni e prototipi di funzioni. Parametri formali e variabili automatiche. Stack di esecuzione.  Passaggio dei parametri per copia e per riferimento. Ricorsione.  Puntatori, indirizzi e gestione della memoria. Aritmetica degli indirizzi. Puntatori generici.  Puntatori a funzioni. Stringhe. Array. Allocazione dinamica di memoria.  Spostamento di dati in memoria. Duplicazione di stringhe.  Tipi aggregati. Strutture. Campi di bit. Union. Definizione di sinonimi per i tipi (typedef).  Input e output. L'input/output standard. Input e output formattato. File. P  rincipali operazioni di accesso ai file.  La libreria standard. Funzioni matematiche.  Modularizzazione del codice.</p>
10	<p>Algoritmi e strutture dati astratte  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.  Stack. Liste. Alberi binari. Alberi binari di ricerca. Alberi n-ari. Ricerca in profondità.  Grafì. Visita di un grafo.</p>
	<b>ESERCITAZIONI TEORICHE</b>
2	Rappresentazione delle informazioni.
2	Algebra booleana
4	Reti combinatorie
4	Reti sequenziali
24	Programmazione strutturata e linguaggio C. Algoritmi e strutture dati.
	<b>ANALISI E DISCUSSIONE DI CASI DI STUDIO</b>
3	Algoritmi di ordinamento
3	Algoritmi ricorsivi
3	Implementazione di liste
3	Ricerca in grafi ed alberi
3	Multiplexer, contatori e altre reti
	<b>PRESENTAZIONI E DISCUSSIONI IN AULA</b>
12	Discussione in aula di progetti, implementazioni preparati durante le esercitazioni di gruppo e degli argomenti assegnati dal docente.
	<b>TESTI CONSIGLIATI</b>
	<ul style="list-style-type: none"> <li>– J. Glenn Brookshear. <b>Informatica. Una panoramica generale.</b> Pearson</li> <li>– Stephen G. Kochan. <b>PROGRAMMARE IN C.</b> Pearson</li> <li>– M. Morris Mano, Charles R. Kime. <b>Reti logiche.</b> Pearson</li> <li>– Paul J. Deitel, Harvey M. Deitel <b>Il linguaggio C Fondamenti e tecniche di programmazione.</b> Pearson</li> <li>– Dennis M. Ritchie, Brian W. Kernighan. <b>Il linguaggio C.</b> Pearson</li> </ul>