



# UNIVERSITÀ DEGLI STUDI DI PALERMO

<b>DIPARTIMENTO</b>	Ingegneria		
<b>ANNO ACCADEMICO OFFERTA</b>	2023/2024		
<b>ANNO ACCADEMICO EROGAZIONE</b>	2023/2024		
<b>CORSO DILAUREA</b>	INGEGNERIA CIBERNETICA		
<b>INSEGNAMENTO</b>	CALCOLATORI ELETTRONICI C.I.		
<b>CODICE INSEGNAMENTO</b>	18794		
<b>MODULI</b>	Si		
<b>NUMERO DI MODULI</b>	2		
<b>SETTORI SCIENTIFICO-DISCIPLINARI</b>	ING-INF/05		
<b>DOCENTE RESPONSABILE</b>	GENTILE ANTONIO	Professore Associato	Univ. di PALERMO
<b>ALTRI DOCENTI</b>	AGATE VINCENZO	Ricercatore a tempo determinato	Univ. di PALERMO
	GENTILE ANTONIO	Professore Associato	Univ. di PALERMO
<b>CFU</b>	12		
<b>PROPEDEUTICITA'</b>			
<b>MUTUAZIONI</b>			
<b>ANNO DI CORSO</b>	1		
<b>PERIODO DELLE LEZIONI</b>	Annuale		
<b>MODALITA' DI FREQUENZA</b>	Facoltativa		
<b>TIPO DI VALUTAZIONE</b>	Voto in trentesimi		
<b>ORARIO DI RICEVIMENTO DEGLI STUDENTI</b>	<b>AGATE VINCENZO</b>		
	Martedì	15:00 - 17:00	Ricevimento in modalita telematica
	<b>GENTILE ANTONIO</b>		
	Venerdì	10:00 - 12:00	Studio del docente presso DINFO, Edificio 6, III pianoDietro prenotazione per email/sito o telefono: 091-238.62603

DOCENTE: Prof. ANTONIO GENTILE

<b>PREREQUISITI</b>	Nessuno
<b>RISULTATI DI APPRENDIMENTO ATTESI</b>	<p>Conoscenza e capacita di comprensione Lo studente acquisira approfondita conoscenza della programmazione strutturata in linguaggio C. Conoscera i principali strumenti di programmazione. Acquisira 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 acquisira conoscenze di base sulle architetture dei calcolatori e sui sistemi operativi Unix-like.</p> <p>Capacita di applicare conoscenza e comprensione Lo studente sara in grado di valutare le possibili soluzioni software a problemi di complessita media e affrontarne l'implementazione utilizzando strumenti e ambienti di sviluppo per la programmazione in linguaggio C in ambienti Unix-like. Sara in grado di affrontare semplici problemi di rappresentazione binaria delle informazioni. Sara in grado di progettare a livello funzionale circuiti logici per la soluzione di semplici problemi.</p> <p>Autonomia di giudizio Lo studente sara in grado di affrontare in autonomia l'analisi, la progettazione e l'implementazione di software utilizzando la programmazione strutturata. Sara in grado di valutare la qualita del software in termini di semplicita, leggibilita, strutturazione, efficienza e riutilizzabilita.</p> <p>Abilita comunicative Lo studente sara in grado di esporre, efficacemente e con proprieta di linguaggio, analisi e soluzioni di problemi affrontabili con la programmazione strutturata e con la progettazione funzionale di circuiti logici, nonche di problemi di rappresentazione delle informazioni.</p> <p>Capacita d'apprendimento Lo studente sara in grado di affrontare in maniera autonoma problemi di programmazione strutturata individuando e integrando soluzioni parziali gia disponibili, sia formalizzate sia implementate. Sara in grado di approfondire in autonomia la conoscenza di moduli software e interfacce di programmazione. Sara in grado di approfondire la conoscenza dei linguaggi e paradigmi di programmazione, dei sistemi operativi, delle architetture dei calcolatori e dei circuiti logici.</p>
<b>VALUTAZIONE DELL'APPRENDIMENTO</b>	<p>Le conoscenze e le competenze acquisite dallo studente saranno verificate attraverso prove finali scritte per i due moduli di Reti Logiche e di Fondamenti di Programmazione. Le prove finali possono essere sostituite da diverse prove scritte parziali nel corso dell'anno. Ogni prova scritta riguardera' degli esercizi che verteranno sui contenuti dei moduli dell'insegnamento. Essa ha lo scopo di verificare le conoscenze dello studente riguardo agli argomenti affrontati durante il corso e la sua capacita' di applicare le conoscenze acquisite. La valutazione della prova terra' conto della capacita' di comunicazione delle scelte implementative da parte dello studente, nonche' la sua autonomia di giudizio nel proporre soluzioni adeguate all'applicazione richiesta.</p> <p>Il voto finale in trentesimi, nell'intervallo 18/30-30/30 con Lode, e' ottenuto come media della valutazione complessiva delle singole prove.</p> <p>In accordo con i descrittori di Dublino, la formulazione delle prove permette una valutazione dei risultati attesi in relazione al voto finale come segue:</p> <ul style="list-style-type: none"><li>- da 18/30 a 20/30: mediocre o sufficiente conoscenza e capacita' di comprensione degli argomenti trattati, parziale capacita' di applicazione delle conoscenze acquisite per la risoluzione dei problemi proposti; parziale autonomia di giudizio, abilita' comunicative e capacita' di apprendere.</li><li>- da 21/30 a 23/30: sufficiente o discreta conoscenza e capacita' di comprensione degli argomenti trattati, sufficiente capacita' di applicazione delle conoscenze acquisite per la risoluzione dei problemi proposti, sufficiente autonomia di giudizio, abilita' comunicative e capacita' di apprendere.</li><li>- da 24/30 a 26/30: discreta conoscenza e capacita' di comprensione degli argomenti trattati, discreta capacita' di applicazione delle conoscenze acquisite per la risoluzione dei problemi proposti, sufficiente autonomia di giudizio, abilita' comunicative e capacita' di apprendere.</li><li>- da 27/30 a 30/30 e lode: buona o eccellente conoscenza e capacita' di comprensione degli argomenti trattati, buona o eccellente capacita' di applicazione delle conoscenze acquisite per la risoluzione dei problemi proposti, buona o eccellente autonomia di giudizio, abilita' comunicative e capacita' di apprendere</li></ul>
<b>ORGANIZZAZIONE DELLA DIDATTICA</b>	Lezioni frontali Esercitazioni in aula e nelle aule informatiche

**MODULO  
RETI LOGICHE**

*Prof. ANTONIO GENTILE*

**TESTI CONSIGLIATI**

1. M. Morris Mano, Charles R. Kime, Reti Logiche, III Edizione italiana, Pearson Education Italia, ISBN: 88-7192-142-9

**TIPO DI ATTIVITA'**

A

**AMBITO**

50283-Matematica, informatica e statistica

**NUMERO DI ORE RISERVATE ALLO STUDIO  
PERSONALE**

96

**NUMERO DI ORE RISERVATE ALLE ATTIVITA'  
DIDATTICHE ASSISTITE**

54

**OBIETTIVI FORMATIVI DEL MODULO**

Lo studente sara' capace di applicare le metodologie studiate in contesti differenti e di apprendere processi di analisi e sintesi relativi ai circuiti logici.

**PROGRAMMA**

<b>ORE</b>	<b>Lezioni</b>
12	Introduzione al corso, concetto di informazione e sua natura logaritmica. Rappresentazione dei numeri interi in una base generica, numeri binari. Conversione di numeri da una base ad un'altra, rappresentazione di numeri in virgola mobile e di numeri negativi. Rappresentazione numerica di immagini e suoni, codice ASCII. Introduzione all'algebra di Boole. Porte logiche, funzioni booleane, logica negata, teoremi e proprieta' dell'algebra, forme normali. Mappe di Karnaugh, minimizzazione di funzione booleane. Sintesi di reti combinatorie. Circuiti integrati (full adder, sommatore, multiplexer, decoder).
16	Circuiti sequenziali; Modelli di Mealy e di Moore; Latch SR, Latch di tipo D; flip flop JK, T, D, SR ; Master slave; flip flop sensibili alle variazioni; Analisi di reti sequenziali sincrone; concetto di Stato, diagramma di stato. Equazioni caratteristiche dei Flip-flop; Procedimenti di sintesi delle reti sequenziali sincrone: Codifica degli stati. Sintesi di reti sequenziale. Sintesi con flip-flop D. Verifica del funzionamento di reti sequenziali, Simulazione di reti sequenziali; Sintesi con flip-flop D, flip-flop T e flip-flop JK. Sintesi di riconoscitori di sequenza, Sintesi di reti sequenziali a partire dal diagramma di stato. Registri, Contatori, ROM, PAL, PLA
8	Architettura di un calcolatore, CPU, ALU, RAM, BUS, macchina di von Neumann. Concetto di programma. Processore. Sistemi CISC, sistemi RISC. Logica cablata, logica microprogrammata. Introduzione ai sistemi operativi
<b>ORE</b>	<b>Esercitazioni</b>
4	Rappresentazione dell'informazione. Algebra Booleana
7	Analisi e Sintesi di reti Combinatorie, Rappresentazione sulla Mappa di Karnaugh; Forme Canoniche Congiuntive e Disgiuntive
7	Analisi e Sintesi di Reti Sequenziali, Riconoscitori di Sequenze

**MODULO  
FONDAMENTI DI INFORMATICA**

*Prof. VINCENZO AGATE*

**TESTI CONSIGLIATI**

1. J. Glenn Brookshear - Stephen G. Kochan, Fondamenti di informatica e programmazione in C, Pearson, ISBN-10: 8865183691  
 2. Brian W. Kernighan , Dennis M. Ritchie, Il linguaggio C - Principi di programmazione e manuale di riferimento, Pearson, ISBN-10: 8891908231  
 3. Paul J. Deitel, Harvey M. Deitel, Maselli, Il linguaggio C, Fondamenti e tecniche di programmazione, ISBN-10: 8891901652, Pearson  
 Per approfondimenti: Igor Zhirkov, Low-Level Programming: C, Assembly, and Program Execution on Intel® 64 Architecture, ISBN-13 (pbk): 978-1-4842-2402-1, ebook disponibile su SpringerLink per gli utenti Unipa  
 Inoltre, Al Aho and Jeff Ullman, Foundations of Computer Science, libro fuori commercio ma reso disponibile online dagli autori al link <http://infolab.stanford.edu/~ullman/focs.html>

<b>TIPO DI ATTIVITA'</b>	A
<b>AMBITO</b>	50283-Matematica, informatica e statistica
<b>NUMERO DI ORE RISERVATE ALLO STUDIO PERSONALE</b>	96
<b>NUMERO DI ORE RISERVATE ALLE ATTIVITA' DIDATTICHE ASSISTITE</b>	54

**OBIETTIVI FORMATIVI DEL MODULO**

Al termine del corso lo studente sara' in grado di valutare, analizzare, comunicare e implementare le possibili soluzioni software a semplici problemi utilizzando l'acquisita padronanza del linguaggio C.

**PROGRAMMA**

<b>ORE</b>	<b>Lezioni</b>
3	Introduzione al corso. Concetto di algoritmo, rappresentazione degli algoritmi, pseudo codice e diagrammi di flusso.
2	Linguaggi di programmazione, prospettiva storica. Compilazione dei programmi. Indipendenza dalla macchina. Concetti della programmazione tradizionale. Unità procedurali. Implementazione del linguaggio. Compilazione del primo programma. Esecuzione del primo programma.
4	Variabili, tipi di dati ed espressioni aritmetiche. Differenza tra variabili e costanti. Assegnamento. Operatori. Priorità tra gli operatori. Introduzione al linguaggio C. Variabili, costanti, tipi, operatori nel linguaggio C.
3	Algoritmi iterativi. Efficienza e correttezza di un algoritmo. Introduzione all'analisi computazionale di un algoritmo: complessità di tempo e di memoria.
5	Programmi iterativi in linguaggio C. Flusso di controllo, strutture iterative (for, while e do-while). Strutture di selezione (if-else) e di selezione tra più alternative (switch-case). Annidamento delle strutture. Comprensione e simulazione di algoritmi/programmi.
4	Funzioni e struttura dei programmi. Argomenti, valori di ritorno e variabili locali. Visibilità e scope di una variabile. Variabili globali, automatiche e statiche. Corpo di una funzione e chiamate a funzioni. Ricorsione. Progettazione modulare di software: tecniche bottom-up e top-down.
8	Il concetto di astrazione. Strutture dati elementari. Vettori, matrici, record, liste, code e pile e loro funzionamento logico. Il puntatore nel linguaggio C. Puntatori e array. Aritmetica dei puntatori. Array multi-dimensionali. Implementazione di strutture e tipi personalizzati. Implementazione di strutture dati elementari. Operazioni di ricerca e di ordinamento in collezioni di dati.
2	Stringhe di caratteri. Utilizzo delle stringhe in C: array di caratteri, stringhe di caratteri di lunghezza variabile, sequenze di escape.
2	Gestione di input e output nel linguaggio C. Gestione di file. Enumerazioni. Preprocessore.

<b>ORE</b>	<b>Esercitazioni</b>
6	Progettazione di semplici algoritmi attraverso diagrammi di flusso. Passaggio da diagrammi di flusso a codice C. Primi programmi in C: compilazione, linking ed esecuzione. Implementazione di programmi per la manipolazione di dati numerici.
3	Progettazione di algoritmi che prevedono la selezione e strutture iterative attraverso i diagrammi di flusso. Implementazione corrispondente in linguaggio C.
3	Esercizi sulla progettazione di software modulare attraverso un uso corretto delle funzioni. Progettazione e implementazione di algoritmi che utilizzano la ricorsione.
6	Progettazione e implementazione di algoritmi che utilizzano stringhe di caratteri, vettori e matrici. Utilizzo dei puntatori. Esercizi sull'implementazione di liste, code e pile.
3	Implementazione di software per l'I/O da file. Preparazione alla prova scritta.