



UNIVERSITÀ DEGLI STUDI DI PALERMO

DIPARTIMENTO	Ingegneria		
ANNO ACCADEMICO OFFERTA	2022/2023		
ANNO ACCADEMICO EROGAZIONE	2022/2023		
CORSO DILAUREA	INGEGNERIA CIBERNETICA		
INSEGNAMENTO	CALCOLATORI ELETTRONICI C.I.		
CODICE INSEGNAMENTO	18794		
MODULI	Si		
NUMERO DI MODULI	2		
SETTORI SCIENTIFICO-DISCIPLINARI	ING-INF/05		
DOCENTE RESPONSABILE	DE PAOLA ALESSANDRA	Professore Associato	Univ. di PALERMO
ALTRI DOCENTI	DE PAOLA ALESSANDRA	Professore Associato	Univ. di PALERMO
	GENTILE ANTONIO	Professore Associato	Univ. di PALERMO
CFU	12		
PROPEDEUTICITA'			
MUTUAZIONI			
ANNO DI CORSO	1		
PERIODO DELLE LEZIONI	Annuale		
MODALITA' DI FREQUENZA	Facoltativa		
TIPO DI VALUTAZIONE	Voto in trentesimi		
ORARIO DI RICEVIMENTO DEGLI STUDENTI	DE PAOLA ALESSANDRA Venerdì 15:00 16:00 stanza del docente, viale delle scienze, ed. 6, prima scala, 3° piano GENTILE ANTONIO 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.ssa ALESSANDRA DE PAOLA

PREREQUISITI	Conoscenze di base di matematica e scienze appresi alla scuola media superiore.
RISULTATI DI APPRENDIMENTO ATTESI	<p>Conoscenza e capacita' di comprensione Lo studente alla fine del corso acquisira' sia una buona conoscenza delle tecniche di rappresentazione delle informazione e della loro elaborazione tramite reti sequenziali e combinatorie, sia una buona conoscenza delle tecniche di ottimizzazione di tali sistemi. Lo studente apprendera' inoltre la sintassi e le tecniche di programmazione in linguaggio C, le strutture dati e gli algoritmi fondamentali.</p> <p>Capacita' di applicare conoscenza e comprensione Lo studente sara' in grado di applicare le tecniche studiate per progettare, a livello logico e funzionale, reti combinatorie e sequenziali e analizzarne il funzionamento. Lo studente sara' in grado di utilizzare strumenti e ambienti di sviluppo per la programmazione in linguaggio C e sapra' implementare soluzioni software.</p> <p>Autonomia di giudizio Lo studente sara' in grado sia di effettuare l'analisi di un problema che di progettare, a partire da una descrizione verbale, una opportuna soluzione software. Sara' in grado di valutare la qualita' di una soluzione software in termini di semplicita', leggibilita', efficienza e possibilita' di riutilizzo. Sara' in grado di capire i principi di funzionamento del calcolatore.</p> <p>Abilita' comunicative Lo studente acquisira' la capacita' di comunicare ed esprimere problematiche inerenti l'oggetto del corso. Sara' in grado di descrivere con terminologia appropriata un circuito logico. Sara' in grado, utilizzando un linguaggio semplice e chiaro, di descrivere i processi di analisi e di sintesi di soluzioni software.</p> <p>Capacita' d'apprendimento Lo studente avra' capacita' di applicazione delle metodologie studiate in contesti differenti e di apprendere processi di analisi e sintesi relativi a programmi software in programmazione strutturata e circuiti logici.</p>
VALUTAZIONE DELL'APPRENDIMENTO	<p>L'apprendimento viene valutato attraverso due prove scritte che insistono sui due moduli del corso ed un eventuale colloquio orale.</p> <p>La prova scritta riguardante gli argomenti del modulo di Reti Logiche consiste prevalentemente nella risposta ad alcuni quesiti a risposta chiusa o aperta e in quesiti di progettazione di circuiti sequenziali e combinatori. I quesiti tenderanno a verificare a) le conoscenze acquisite; b) la capacita' di progettare circuiti che rispondano ad alcune specifiche, c) la capacita' di organizzare ed esporre le conoscenze tecniche. Il punteggio massimo si ottiene se la verifica accerta il pieno possesso dei tre seguenti aspetti: capacita' di svolgere i quesiti riguardanti la rappresentazione dell'informazione, la capacita' di progettare o ottimizzare circuiti logici che assolvono a specifici compiti, la capacita' di descrivere e confrontare diverse soluzioni circuitali.</p> <p>La prova scritta riguardante gli argomenti del modulo di Fondamenti di Informatica sarà costituita da esercizi di programmazione, relativi alla generazione di uno o piu programmi in linguaggio C, in base ad alcune specifiche tecniche date, e quesiti a risposta aperta. La prova tendera' a verificare la conoscenza del linguaggio di programmazione C, la capacita' di trovare soluzioni a semplici problemi tipici della disciplina, la capacita' di creare software funzionanti, la conoscenza degli argomenti affrontati durante il corso.</p> <p>Il punteggio massimo si ottiene se la verifica accerta il pieno possesso dei seguenti aspetti: l'utilizzo di costrutti del linguaggio in modo sintatticamente corretto; la capacita' di comporre i costrutti per risolvere problemi riguardanti l'acquisizione, l'elaborazione e la memorizzazione di informazioni; la creazione di software che possano essere eseguiti senza presentare malfunzionamenti; la piena conoscenza degli argomenti del corso. Durante il colloquio orale lo studente dovra' essere in grado di discutere le soluzioni proposte durante le prove scritte.</p> <p>Il voto finale sarà calcolato come media delle valutazioni delle 3 prove. La valutazione generale si basa sui seguenti criteri di massima: a) eccellente (30 - 30 e lode): ottima conoscenza degli argomenti, ottima proprieta' di linguaggio, buona capacita' analitica, lo studente e' in grado di applicare le conoscenze per risolvere i problemi proposti; b) molto buono (26 - 29): buona padronanza degli argomenti, piena proprieta' di linguaggio, lo studente e' in grado di applicare le conoscenze per risolvere i problemi proposti;</p>

	<p>c) buono (24 - 25): conoscenza di base dei principali argomenti, discreta proprietà di linguaggio con limitata capacità di applicare autonomamente le conoscenze alla soluzione dei problemi proposti;</p> <p>d) discreto (21 - 23): limitata padronanza degli argomenti principali dell'insegnamento, discreta proprietà di linguaggio, discreta capacità di applicare autonomamente le conoscenze acquisite;</p> <p>e) sufficiente (18 - 20): minima conoscenza di base degli argomenti principali dell'insegnamento e del linguaggio tecnico, scarsa capacità di applicare autonomamente le conoscenze acquisite</p> <p>f) insufficiente: non possiede una conoscenza minima accettabile dei contenuti degli argomenti trattati nell'insegnamento.</p>
ORGANIZZAZIONE DELLA DIDATTICA	Lezioni frontali, esercitazioni in aula ed in laboratorio.

MODULO RETI LOGICHE

Prof. ANTONIO GENTILE

TESTI CONSIGLIATI

1. M. Morris Mano, Charles R. Kime, Reti Logiche, Edizione italiana, Pearson Education Italia, ISBN: 88-7192-142-9
2. J. Glenn Brookshear - Stephen G. Kochan, Fondamenti di informatica e programmazione in C, Pearson, ISBN-10: 8865183691

Per approfondimenti: Brock J. LaMeres, Introduction to logic circuits & logic design with Verilog, Springer, ISBN 978-3-319-53882-2, disponibile online su SpringerLink per gli utenti Unipa

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 sarà capace di applicare le metodologie studiate in contesti differenti e di apprendere processi di analisi e sintesi relativi ai circuiti logici.

PROGRAMMA

ORE	Lezioni
12	<p>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.</p> <p>Introduzione all'algebra di Boole. Porte logiche, funzioni booleane, logica negata, teoremi e proprietà dell'algebra, forme normali. Mappe di Karnaugh, minimizzazione di funzione booleane. Sintesi di reti combinatorie. Circuiti integrati (full adder, sommatore, multiplexer, decoder).</p>
16	<p>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</p>
8	<p>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</p>

ORE	Esercitazioni
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.ssa ALESSANDRA DE PAOLA

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>

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

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

ORE	Lezioni
3	Introduzione al corso. Elaborazione dei dati, architettura dei computer, linguaggio macchina, esecuzione dei programmi, istruzioni aritmetico/logiche. 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.

ORE	Esercitazioni
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.