



UNIVERSITÀ DEGLI STUDI DI PALERMO

DIPARTIMENTO	Ingegneria
ANNO ACCADEMICO OFFERTA	2023/2024
ANNO ACCADEMICO EROGAZIONE	2023/2024
CORSO DILAUREA	INGEGNERIA DELL'INNOVAZIONE PER LE IMPRESE DIGITALI
INSEGNAMENTO	CALCOLATORI ELETTRONICI C.I.
CODICE INSEGNAMENTO	18794
MODULI	Si
NUMERO DI MODULI	2
SETTORI SCIENTIFICO-DISCIPLINARI	ING-INF/05
DOCENTE RESPONSABILE	GALLO PIERLUIGI Professore Associato Univ. di PALERMO
ALTRI DOCENTI	GALLO PIERLUIGI 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	GALLO PIERLUIGI Venerdì 15:00 17:00 Ufficio del docente

PREREQUISITI	Uso di base del calcolatore con sistema operativo Windows/Linux/macOS: copia, incolla e rinomina di files, installazione di applicazioni, uso dell'interfaccia grafica.
RISULTATI DI APPRENDIMENTO ATTESI	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. 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. 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. 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 ed efficienza. 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. 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.
VALUTAZIONE DELL'APPRENDIMENTO	La valutazione dell'apprendimento avviene mediante una prova in itinere per il modulo di Reti Logiche e lo sviluppo di codice da scrivere in linguaggio C secondo un tema assegnato dal docente per il modulo di Fondamenti di Programmazione. La prova in itinere consiste nella compilazione di un test a risposta multipla riguardante la parte di reti logiche. Il questionario comprende quindici quesiti a risposte multiple sugli argomenti relativi alle reti logiche, alla struttura dei calcolatori elettronici e alla rappresentazione dell'informazione a basso livello. La risposta a ogni quesito viene valutata con un punteggio pari a 2, se la risposta e' corretta, 0 se la risposta non e' fornita, -1 se la risposta e' errata. Il voto della prova, in trentesimi, e' ottenuto sommando i punteggi di tutte le risposte ai quesiti. La prova in itinere e' ritenuta superata con voto uguale o superiore a 18/30. Per la prova del modulo di Fondamenti di Programmazione, deve essere scritto il codice sorgente in linguaggio C secondo un testo assegnato dal docente. Il codice deve essere in grado di essere compilato e l'eseguibile deve essere funzionante. Il codice e' valutato dal docente secondo le seguenti caratteristiche: modularita' del codice, indentazione, presentazione del risultato, adattivita' a dati della stessa natura ed aderenza al problema proposto. Gli studenti non in corso o che non abbiano superato la prova in itinere, sono tenuti a sostenere una prova complessiva integrante i due moduli, con le medesime modalita', rispettivamente, della prova in itinere e della prova finale al calcolatore. I voti della prova in itinere sul modulo di reti logiche e di quella di Fondamenti di programmazione sono sommati con pesi rispettivamente di 0.3 e 0.7 (per un voto di Fondamenti di Programmazione > 18/30) per ottenere una valutazione complessiva in trentesimi; altrimenti, varra' il voto di Fondamenti di Programmazione per tutta la materia. Il voto finale in trentesimi, nell'intervallo 18/30-30/30 con Lode, e' ottenuto mediante media delle valutazioni della prova di Reti Logiche e di quella di Fondamenti di Programmazione. La lode e' assegnata agli studenti che abbiano ottenuto 30/30 per entrambe le prove e che rispondano correttamente ad una domanda orale. La formulazione delle prove fornisce una valutazione dei risultati attesi in relazione al voto finale come segue: -da 18/30 a 20/30: sufficiente conoscenza e capacita' di comprensione degli argomenti trattati, capacita' di applicazione delle conoscenze acquisite per la

	risoluzione dei problemi proposti, autonomia di giudizio, abilità comunicative e capacità di apprendere. -da 21/30 a 23/30: discreta conoscenza e capacità di comprensione degli argomenti trattati, capacità di applicazione delle conoscenze acquisite per la risoluzione dei problemi proposti, autonomia di giudizio, abilità comunicative e capacità di apprendere. -da 24/30 a 26/30: buona conoscenza e capacità di comprensione degli argomenti trattati, capacità di applicazione delle conoscenze acquisite per la risoluzione dei problemi proposti, autonomia di giudizio, abilità comunicative e capacità di apprendere. -da 27/30 a 30/30 e lode: eccellente conoscenza e capacità di comprensione degli argomenti trattati, capacità di applicazione delle conoscenze acquisite per la risoluzione dei problemi proposti, autonomia di giudizio, abilità comunicative e capacità di apprendere.
ORGANIZZAZIONE DELLA DIDATTICA	Le lezioni sono tenute in aula dal docente con supporto di video-proiezione del desktop del computer del docente. Si potrà portare in aula il proprio notebook per provare il codice mostrato in aula.

MODULO ARCHITETTURA DEI CALCOLATORI

Prof. PIERLUIGI GALLO

TESTI CONSIGLIATI

- M. Morris Mano - Charles Kime - Tom Martin . Reti logiche 5/Ed. Italiana ISBN: 9788891905819. Pearson Editore - Slides proiettate durante la lezione.

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

Il modulo si propone di fornire agli studenti i concetti di base riguardanti: progettazione dei circuiti logici e delle reti sequenziali, architettura del calcolatore, struttura e funzionamento del microprocessore.

PROGRAMMA

ORE	Lezioni
4	Introduzione al corso. Modalità d'esame. Rappresentazione interna ed esterna. Il segnale analogico vs segnale digitale.
6	Conversione da base 10 a base 2. Conversione di numeri frazionari. Somme e sottrazioni in base 2. Codici numerici: BCD, ASCII, Gray. Standard IEEE 754. Rappresentazione di immagini e suoni.
3	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.
3	Mappe di Karnaugh. Implicanti, primi implicanti e primi implicanti essenziali di funzioni booleane. Minimizzazione di funzioni booleane. Operatore XOR. Operatori funzionalmente completi.
3	Reti combinatorie. Decoder ed encoder. Espansione in serie di decoder. Encoder con priorità. Multiplexer e demultiplexer.
3	Reti combinatorie. Sintesi con decoder. Sintesi con multiplexer. Half-adder e Full-adder.
6	Reti sequenziali. Latch, Flip-Flop, contatori e registri.
3	Architettura di von Neumann. CPU e memoria RAM. Memoria ROM. Memorie di massa. Cache.
3	Struttura di un microprocessore. Segmentazione della memoria.
12	Modelli di memoria. Metodi di indirizzamento. Offset. Vettori. Interrupt. LEA. Subroutines. Addizione ad 8 e 16 bit. Stack. Moltiplicazione Salti condizionati e non condizionati. Cicli. Scorrimento e rotazione.
3	Logica e Diagrammi di flusso. Teorema di Bohm e Jacopini.
5	Il compilatore, file sorgente, eseguibile, ottimizzazione, ambienti di sviluppo, make e Makefile.

MODULO FONDAMENTI DI PROGRAMMAZIONE

Prof. PIERLUIGI GALLO

TESTI CONSIGLIATI

-J. Glenn Brookshear, Stephen G. Kochan, "Fondamenti di informatica e programmazione in C", Pearson
 -J. R. Hanly, E. B. Koffman, "Problem solving e programmazione in C" Apogeo
 -Slides proiettate durante la lezione.

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

Il modulo si propone di fornire agli studenti i concetti di base della programmazione in linguaggio C. Gli studenti saranno in grado di analizzare, comunicare e implementare le possibili soluzioni software a problemi applicativi utilizzando l'acquisita padronanza del linguaggio C.

PROGRAMMA

ORE	Lezioni
2	Presentazione del modulo di programmazione. Modalita' d'esame. Descrizione del programma in C "Hello World!".
2	Printf. Tipi da dato. Formattazione della stringa di visualizzazione per printf. Commenti nel codice. Conversione da un tipo all'altro: intrinseco e casting.
3	Incremento di una variabile. Sintassi del ciclo for. Somma da 1 a n tramite cicli in C. Ciclo for per conteggio alla rovescia. Algoritmo per la somma parziale di Gauss. Significato di pseudocodice. Definizione di euristica ed algoritmo. Cicli for annidati. Operatori relazionali.
3	Ciclo while a valutazione anticipata. Ciclo do...while a valutazione posticipata. Cicli while w do...while annidati. Diagrammi di flusso. Pericolo del loop infinito. I costrutti if...else e switch...case.
3	Operatori booleani per espressioni. Tipo booleano. Operatori booleani bit a bit. Introduzione agli array monodimensionali. Caratteri accentati e speciali della printf. Macro del preprocessore. Operatore ternario.
3	Array monodimensionale. Inizializzazione dell'array per enumerazione e tramite ciclo. Array con/ senza size. Esempi nell'uso dell'array. Qualificatore const.
3	Prodotto scalare tra vettori. Array bidimensionali. Linearizzazione delle matrici anche con macro. Row major e column major. Buone cattive pratiche di programmazione con gli array. Copia da vettore a matrice.
3	Swap tra variabili. Trasposta di una matrice su una nuova matrice risultato e sulla stessa matrice. Prodotto matriciale. Introduzione alle funzioni in C.
3	Variabili locali. Esempi con variabili locali. Parametri formali e attuali. Vantaggi nell'uso delle funzioni e differenza con le macro con parametri. Valori ritornati dalla funzione. Passaggio di variabile ad una per valore (per copia).
3	Funzioni che chiamano funzioni. Passaggio di variabile per indirizzo. Passaggio per indirizzo degli array per default. Minimo/massimo di un vettore. Ordinamento per selezione di un vettore. Prototipo di funzione.
2	Variabili globali. Algoritmi ricorsivi.
3	Struct. Struct globale. Passaggio di una struct di default. Funzioni che chiamano funzioni contenenti struct.
2	Struct di array. Array di struct. Introduzione alle stringhe di caratteri.
3	Merge e conteggio parole nelle stringhe di caratteri. Input da tastiera con getchar. Conversione da stringa a intero.
3	Libreria String.h. Ordine di complessita' degli algoritmi: caso lineare. Ricerca sequenziale.
3	Ordine di complessita' degli algoritmi: caso logaritmico per la ricerca binaria, caso quadratico con l'ordinamento per selezione.
2	Gestione dei files. Stdin-Stdout-Stderr. Introduzione ai puntatori.
3	Scambio di valori e di indirizzi tra variabili tramite puntatori. Passaggio di una variabile ad una funzione per indirizzo. Passaggio di una struct per indirizzo tramite puntatori.
2	Array dinamici
3	Liste