



UNIVERSITÀ DEGLI STUDI DI PALERMO

DIPARTIMENTO	Ingegneria		
ANNO ACCADEMICO OFFERTA	2020/2021		
ANNO ACCADEMICO EROGAZIONE	2020/2021		
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	GAMBINO ORAZIO	Ricercatore	Univ. di PALERMO
ALTRI DOCENTI	GAMBINO ORAZIO	Ricercatore	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	GAMBINO ORAZIO Lunedì 10:00 12:00 Chat di Teams, previo appuntamento concordato via email. Martedì 10:00 12:00 Chat di Teams, previo appuntamento concordato via email.		

DOCENTE: Prof. ORAZIO GAMBINO

PREREQUISITI	Uso di base del calcolatore con sistema operativo Windows: copia, incolla e rinomina di files, installazione di applicazioni, uso dell'interfaccia grafica.
RISULTATI DI APPRENDIMENTO ATTESI	<p>Conoscenza e capacita' di comprensione</p> <p>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.</p> <p>Capacita' di applicare conoscenza e comprensione</p> <p>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.</p> <p>Autonomia di giudizio</p> <p>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.</p> <p>Abilita' comunicative</p> <p>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</p> <p>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>
VALUTAZIONE DELL'APPRENDIMENTO	<p>La valutazione dell'apprendimento avviene mediante una prova in itinere per il modulo di Reti Logiche e un progetto finale (tesina) assegnato dal docente da scrivere in linguaggio C con domande riguardanti la programmazione in C per il modulo di Fondamenti di Programmazione.</p> <p>La prova in itinere consiste nella compilazione di un test a risposta multipla riguardante la parte di reti logiche e programmazione assembly.</p> <p>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.</p> <p>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.</p> <p>Per la prova del modulo di Fondamenti di Programmazione, deve essere creato un programma in linguaggio C da parte di un gruppo di 2 o 3 allievi secondo un tema lasciato dal docente. Insieme al programma, dovra' essere consegnata una documentazione relativa al progetto secondo quanto spiegato durante il corso.</p> <p>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.</p> <p>I voti della prova in itinere sul modulo di reti logiche e di quella del progetto sono sommati con pesi rispettivamente di 0.3 e 0.7 per ottenere una valutazione complessiva in trentesimi.</p>

	<p>La discussione del progetto vertera' sulla realizzazione del programma e sugli argomenti trattati nel corso sulla programmazione C. 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.</p> <p>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, abilita' comunicative e capacita' di apprendere. - da 21/30 a 23/30: discreta conoscenza e capacita' di comprensione degli argomenti trattati, capacita' di applicazione delle conoscenze acquisite per la risoluzione dei problemi proposti, autonomia di giudizio, abilita' comunicative e capacita' di apprendere. - da 24/30 a 26/30: buona conoscenza e capacita' di comprensione degli argomenti trattati, capacita' di applicazione delle conoscenze acquisite per la risoluzione dei problemi proposti, autonomia di giudizio, abilita' comunicative e capacita' di apprendere. - da 27/30 a 30/30 e lode: eccellente conoscenza e capacita' di comprensione degli argomenti trattati, capacita' di applicazione delle conoscenze acquisite per la risoluzione dei problemi proposti, autonomia di giudizio, abilita' comunicative e capacita' di apprendere.</p>
ORGANIZZAZIONE DELLA DIDATTICA	<p>Le lezioni saranno 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. In casi eccezionali, per motivi di emergenza sanitaria o altri tipi di emergenza e dietro l'approvazione degli organi di Ateneo, le lezioni verranno svolte offline tramite pubblicazione delle slides con commento sonoro da parte del docente che si metterà online a disposizione per chiarimenti e delucidazioni sulle lezioni.</p>

MODULO RETI LOGICHE

Prof. ORAZIO GAMBINO

TESTI CONSIGLIATI

- J. Glenn Brookshear, Stephen G. Kochan, "Fondamenti di informatica e programmazione in C", Pearson
- M. Morris Mano, Charles R. Kime, "Reti Logiche", Pearson
- Materiale didattico fornito dal docente (Slide delle lezioni ed esercizi)

TIPO DI ATTIVITA'	A
AMBITO	50283-Matematica, informatica e statistica
NUMERO DI ORE RISERVATE ALLO STUDIO PERSONALE	102
NUMERO DI ORE RISERVATE ALLE ATTIVITA' DIDATTICHE ASSISTITE	48

OBIETTIVI FORMATIVI DEL MODULO

Il modulo si propone di fornire agli studenti i concetti di base sull'algebra di Boole e sulle reti logiche, e sulla rappresentazione e l'elaborazione delle informazioni dal livello del bit fino alle strutture dati astratte.

Gli studenti acquisiranno una conoscenza di base delle problematiche inerenti le metodologie di progettazione di reti logiche combinatorie e sequenziali e di algoritmi e strutture dati.

PROGRAMMA

ORE	Lezioni
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.
4	Rappresentazione delle informazioni. Rappresentazione in complemento alla base. Rappresentazione di numeri reali: virgola fissa, virgola mobile. Rappresentazione di caratteri alfanumerici. Codice ASCII. Rappresentazione di immagini.
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	Algebra Booleana. Mappe di Karnaugh. Implicanti, primi implicanti e primi implicanti essenziali di funzioni booleane. Minimizzazione di funzioni booleane. Operatore XOR. Operatori funzionalmente completi.
2	Reti combinatorie. Decoder ed encoder. Espansione in serie di decoder. Encoder con priorità. Multiplexer e demultiplexer.
2	Reti combinatorie. Sintesi con decoder. Sintesi con multiplexer. Half-adder e Full-adder. ALU 74181
4	Reti sequenziali. Latch. Flip-Flop.
2	Reti sequenziali: contatori e registri.
5	Architettura di von Neumann. CPU e memoria RAM. Memoria ROM. Memorie di massa. Cache.
4	Struttura del microprocessore Intel 8086. Descrizione e funzionamento delle unità BIU ed EU. Registri del microprocessore 8086. Segmentazione della memoria.
12	Programmazione Assembly 8086. Struttura di un programma assembly. 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.
2	Presentazione del modulo di programmazione. 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.
2	Pre-/Post- incremento di una variabile. Sintassi del ciclo for. Somma da 1 a 13 tramite cicli sia in asm8086 che 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. Teorema di Bohm e Jacopini. La selezione: il costrutto if...else. Costrutto switch...case.
ORE	Esercitazioni
6	Simulazione prova d'esame

**MODULO
FONDAMENTI DI PROGRAMMAZIONE**

Prof. ORAZIO GAMBINO

TESTI CONSIGLIATI

- J. Glenn Brookshear, Stephen G. Kochan, "Fondamenti di informatica e programmazione in C", Pearson
- B. W. Kernighan, D. M. Ritchie, "Linguaggio C", Pearson

TIPO DI ATTIVITA'	A
AMBITO	50283-Matematica, informatica e statistica
NUMERO DI ORE RISERVATE ALLO STUDIO PERSONALE	102
NUMERO DI ORE RISERVATE ALLE ATTIVITA' DIDATTICHE ASSISTITE	48

OBIETTIVI FORMATIVI DEL MODULO

Il modulo si propone di fornire agli studenti i concetti di base della programmazione strutturata 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
3	#define. Operatore ternario. Set di caratteri italiano. Macro. Operatori logici ed operatori bit a bit. Array unidimensionale. Indicizzazione di un array 1D. Somma di un vettore sia in asm8086 che linguaggio C. Esempi su cicli e vettori. Qualificatore const. Ciclo for con aggiornamento multiplo.
3	Linearizzazione di una matrice. Array 2D. Scambio di valori tra variabili. Trasposta di una matrice. Algoritmo di eliminazione di Gauss-Jordan. Prodotto di matrici. Le funzioni. Passaggio per valore ad una funzione. Variabile locale.
3	Funzioni che chiamano funzioni. Minimo di un array. Passaggio di un array by-reference. Ordinamento per selezione. Variabile globale. Qualificatore static. Ricorsione: fattoriale e Torri di Hanoi.
3	Strutture dati struct e stringhe.
3	Calcolo dell'efficienza di un algoritmo. Ricerca binaria. Visita di un grafo in profondita.
3	Immissione dati da file. Scrittura di un file. Formato grafico .ppm . Impostazione della console di Windows.
3	Semplice elaborazione di file d'immagine, manipolazione degli attributi della console di Windows, esempi di schermate per il gioco del solitario, stdin-stdout-stderr, simboli speciali per infinito e indeterminazione, errore sui floating point, cenni di opengl per il tracciare un grafico di funzione, puntatori, passaggio per riferimento delle variabili.
3	Passaggio per riferimento di una struct, struct con puntatori, sizeof, calloc e malloc, vettore dinamico, swap universale, matrici tramite vettore dinamico, matrice dinamica tramite puntatore doppio, puntatori a void*.
3	Copia universale, realloc, puntatore doppio, matrice triangolare, prodotto tra matrici dinamiche, documentazione per le tesine: grafo delle chiamate, descrizione delle funzioni e descrizione dell'algoritmo. Il gioco del Tris. Enum
3	Liste con strutture statiche, lista dinamica, inserimento in lista di un elemento, cancellazione dalla lista di un elemento, immissione dati tramite lista.
3	Lista bidirezionale. Lista ciclica bidirezionale. Lista di liste.
3	Alberi binari. Funzioni relative agli alberi binari con puntatori. Path assoluto e relativo.
3	Uso dei puntatori. Puntatore doppio ad una matrice statica. Puntatore singolo ad una matrice statica. Vettori e matrici eterogenee. Un algoritmo di shuffle. Visualizzazione dei caratteri unicode dei semi dei caratteri unicode. Copia da matrice statica a matrice dinamica.
3	Puntatore a funzione. Nota sull'uso delle variabili globali. Ricognizione dello stato di avanzamento dei progetti degli allievi.
ORE	Esercitazioni
3	Ricognizione dello stato di avanzamento dei progetti degli studenti
3	Ricognizione dello stato di avanzamento dei progetti degli studenti
3	Ricognizione dello stato di avanzamento dei progetti degli student
3	Esempi di programmi svolti dal docente