



UNIVERSITÀ DEGLI STUDI DI PALERMO

| | | | |
|---|--|-------------|------------------|
| DIPARTIMENTO | Ingegneria | | |
| ANNO ACCADEMICO OFFERTA | 2017/2018 | | |
| ANNO ACCADEMICO EROGAZIONE | 2017/2018 | | |
| 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 | PERI DANIELE | Ricercatore | Univ. di PALERMO |
| ALTRI DOCENTI | PERI DANIELE | 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 | PERI DANIELE Mercoledì 15:00 16:00 Ricevimento in modalita a distanza sulla piattaforma MS Teams | | |

| | |
|--|---|
| PREREQUISITI | Nessuno. |
| RISULTATI DI APPRENDIMENTO ATTESI | <p>Conoscenza e capacita' di comprensione Lo studente acquisira' approfondita conoscenza della programmazione strutturata in linguaggio C. Conoscera' i principali strumenti di programmazione in ambiente Unix-like. 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 ed efficienza.</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> |
| VALUTAZIONE DELL'APPRENDIMENTO | <p>La valutazione dell'apprendimento avviene mediante una prova scritta in itinere, una prova scritta finale e una prova orale.</p> <p>La prova scritta in itinere consiste nella compilazione di un questionario comprendente 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, o -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.</p> <p>La prova scritta finale richiede la soluzione, a risposta libera, di otto problemi di programmazione in Linguaggio C che concorrono alla costruzione di un programma applicativo completo e di due esercizi di implementazione, sempre in Linguaggio C, di algoritmi e strutture dati astratte. La risposta a ogni quesito viene valutata con un punteggio da 0, nel caso di soluzione non fornita, a 5, corrispondente a una soluzione priva di errori. Il voto della prova, in trentesimi, e' ottenuto mediante somma pesata dei punteggi di tutte le risposte ai quesiti. Gli studenti non in corso o che non abbiano superato la prova in itinere, sono tenuti a sostenere una prova scritta complessiva integrante due prove, con le medesime modalita, rispettivamente, della prova scritta in itinere e della prova scritta finale.</p> <p>I voti delle due prove scritte, nel caso sia di prova scritta in itinere e prova scritta finale sia di prova scritta complessiva, sono sommati con pesi rispettivamente di 1/4 e 3/4 per ottenere una valutazione complessiva in trentesimi.</p> <p>La prova orale consiste nella discussione della prova scritta finale o di quella complessiva e nell'approfondimento degli argomenti del corso.</p> <p>Il voto finale in trentesimi, nell'intervallo 18/30-30/30 con Lode, e' ottenuto mediante media della valutazione complessiva della prova scritta e di quella orale. La formulazione delle prove fornisce una valutazione dei risultati attesi in relazione al voto finale come segue:</p> <ul style="list-style-type: none"> - 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 |

| | |
|---------------------------------------|---|
| | <p>argomenti trattati, capacita' di applicazione delle conoscenze acquisite per la risoluzione dei problemi proposti, autonomia di giudizio, abilita' comunicative e capacita' di apprendere.</p> <p>- 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.</p> <p>- 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 | Lezioni ed esercitazioni in aula informatica. |

MODULO RETI LOGICHE

Prof. DANIELE PERI

TESTI CONSIGLIATI

- M. Morris Mano, Charles R. Kime, "Reti Logiche", Pearson
- J. Glenn Brookshear, Stephen G. Kochan, "Fondamenti di informatica e programmazione in C", Pearson

| | |
|--|--|
| 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 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. Sistema di numerazione esadecimale. Codici. Codice BCD. Rappresentazione di interi con segno. |
| 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. |
| 4 | Reti sequenziali. Modelli di Mealy e Moore. Latch. Flip-Flop. |
| 2 | Reti sequenziali. Sintesi di reti sequenziali sincrone. |
| 2 | Algoritmi e strutture dati. Ricerca di valori in un array: ricerca lineare e ricerca dicotomica. Algoritmi di ordinamento basati sui confronti: ordinamento per selezione, ordinamento per inserimento, mergesort. Algoritmi di ordinamento non basati sui confronti: Integer sort. |
| 2 | Algoritmi e strutture dati. Stack e espressioni in notazione polacca inversa (RPN). |
| 2 | Algoritmi e strutture dati. Operazioni di base su liste concatenate: creazione di un elemento, aggiunta in testa, scansione della lista. |
| 2 | Algoritmi e strutture dati. Operazioni di base su liste: concatenazione, aggiunta in coda, cancellazione. Operazioni avanzate su liste: inversione, ordinamento, map. |
| 2 | Algoritmi e strutture dati. Alberi binari. Alberi binari di ricerca. Alberi n-ari. Ricerca in profondità. |
| 2 | Algoritmi e strutture dati. Grafi. Visite di grafi. |

| ORE | Esercitazioni |
|-----|---|
| 2 | Rappresentazione delle informazioni. Algebra booleana. |
| 2 | Reti combinatorie. |
| 2 | Reti sequenziali. |
| 2 | Algoritmi e strutture dati. 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. |
| 2 | Algoritmi e strutture dati. Stack. Stack ed espressioni in notazione polacca inversa (RPN). |
| 2 | Algoritmi e strutture dati. Operazioni di base su liste concatenate: creazione di un elemento, aggiunta in testa, scansione della lista. |
| 2 | Algoritmi e strutture dati. Operazioni di base su liste: concatenazione, aggiunta in coda, cancellazione. Operazioni avanzate su liste: inversione, ordinamento, map. |
| 2 | Algoritmi e strutture dati. Alberi binari. Alberi binari di ricerca. Alberi n-ari. Ricerca in profondità. |

**MODULO
FONDAMENTI DI PROGRAMMAZIONE**

Prof. DANIELE PERI

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 | 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 necessari alla comprensione della struttura dei calcolatori elettronici digitali programmabili e ai sistemi operativi Unix-like. 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 | Calcolatori elettronici digitali programmabili. Struttura di un calcolatore. Modello Von Neumann: CPU, memoria, input/output. Microprocessori. |
| 2 | Software di base e software applicativo. Sistemi operativi. Astrazione dall'hardware. Driver. File system. Processi. Sistemi Unix-like. Interfacce a riga di comando, terminale e shell. |
| 2 | Linguaggi di programmazione e sviluppo del software. Linguaggio macchina. Compilazione e interpretazione. Codice sorgente, codice oggetto e codice eseguibile. Compilazione: preprocessore, compilatore, assembler e linker. Moduli e librerie. Caricamento in memoria ed esecuzione dei programmi. Argomenti della riga di comando e codice di uscita. |
| 2 | Programmazione strutturata. Variabili e tipi di dati. Strutture dati. Costanti e literal. Istruzioni di assegnamento. Istruzioni di controllo. Procedure e funzioni. |
| 2 | Astrazioni sui dati. Strutture dati elementari. Array. Liste, stack e code. Alberi. Strutture statiche e dinamiche. Tipi di dati definiti dall'utente. Tipi di dati astratti. |
| 2 | 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. Input e output. Input/output standard. Input e output formattato. Libreria standard. |
| 2 | Programmazione strutturata e linguaggio C. Espressioni. Operatori aritmetici. Operatori relazionali e logici. Operatori bit a bit. Conversioni di tipo. Espressioni condizionali. Precedenza e ordine di valutazione. |
| 2 | Programmazione strutturata e linguaggio C. Operatori di incremento e decremento. Cicli iterativi: for, while e do-while. Istruzioni per l'interruzione e la continuazione dei cicli. |
| 2 | Programmazione strutturata e linguaggio C. Strutture di controllo. Costrutto if-else. Costrutto switch. |
| 2 | Programmazione strutturata e linguaggio C. Definizioni e prototipi di funzioni. Parametri formali e variabili automatiche. Stack di esecuzione. Passaggio dei parametri per copia e per riferimento. Ricorsione. |
| 2 | Programmazione strutturata e linguaggio C. Puntatori, indirizzi e gestione della memoria. Aritmetica degli indirizzi. Puntatori a puntatori. Puntatori generici. |
| 2 | Programmazione strutturata e linguaggio C. Stringhe. Array. Allocazione dinamica di memoria. Spostamento di dati in memoria. Duplicazione di stringhe. |
| 2 | Programmazione strutturata e linguaggio C. Tipi aggregati. Strutture. Campi di bit. Union. Definizione di sinonimi per i tipi (typedef). |
| 2 | Programmazione strutturata e linguaggio C. File. Principali operazioni di accesso ai file. |
| 2 | Programmazione strutturata e linguaggio C. Preprocessore. File header. Compilazione condizionale. |
| 2 | Programmazione strutturata e linguaggio C. La libreria standard. Funzioni matematiche. |
| 2 | Programmazione strutturata e linguaggio C. Puntatori a funzioni. |

| ORE | Esercitazioni |
|-----|---|
| 2 | 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. Input e output. Input/output standard. Input e output formattato. Libreria standard. |
| 2 | Programmazione strutturata e linguaggio C. Espressioni. Operatori aritmetici. Operatori relazionali e logici. Operatori bit a bit. Conversioni di tipo. Espressioni condizionali. Precedenza e ordine di valutazione. |

| | |
|---|--|
| 2 | Programmazione strutturata e linguaggio C. Operatori di incremento e decremento. Cicli iterativi: for, while e do-while. Istruzioni per l'interruzione e la continuazione dei cicli. |
| 2 | Programmazione strutturata e linguaggio C. Strutture di controllo. Costrutto if-else. Costrutto switch. |
| 2 | Programmazione strutturata e linguaggio C. Definizioni e prototipi di funzioni. Parametri formali e variabili automatiche. Stack di esecuzione. Passaggio dei parametri per copia e per riferimento. Ricorsione. |
| 2 | Programmazione strutturata e linguaggio C. Puntatori, indirizzi e gestione della memoria. Aritmetica degli indirizzi. Puntatori a puntatori. Puntatori generici. Stringhe. Array. Allocazione dinamica di memoria. Spostamento di dati in memoria. Duplicazione di stringhe. |
| 2 | Programmazione strutturata e linguaggio C. Tipi aggregati. Strutture. Campi di bit. Union. Definizione di sinonimi per i tipi (typedef). |
| 2 | Programmazione strutturata e linguaggio C. File. Principali operazioni di accesso ai file. |
| 2 | Programmazione strutturata e linguaggio C. Preprocessore. File header. Compilazione condizionale. La libreria standard. Funzioni matematiche. |
| 2 | Programmazione strutturata e linguaggio C. Puntatori a funzioni. |