UNIVERSITÀ DEGLI STUDI DI PALERMO

DEPARTMENT	Ingegneria		
ACADEMIC YEAR	2015/2016		
BACHELOR'S DEGREE (BSC)	COMPUTER AND TELECOMMUNICATION ENGINEERING		
INTEGRATED COURSE	COMPUTER FUNDAMENTALS - INTEGRATED COURSE		
CODE	18073		
MODULES	Yes		
NUMBER OF MODULES	2		
SCIENTIFIC SECTOR(S)	ING-INF/05		
HEAD PROFESSOR(S)	GENTILE ANTONIO	Professore Associato	Univ. di PALERMO
OTHER PROFESSOR(S)	VASSALLO GIORGIO GENTILE ANTONIO	Ricercatore Professore Associato	Univ. di PALERMO Univ. di PALERMO
CREDITS	18		
PROPAEDEUTICAL SUBJECTS			
MUTUALIZATION			
YEAR	1		
TERM (SEMESTER)	1° semester		
ATTENDANCE	Not mandatory		
EVALUATION	Out of 30		
TEACHER OFFICE HOURS	**GENTILE ANTONIO** Friday 10:00 12:00 Studio del docente presso DINFO, Edificio 6, III pianoDietro prenotazione per email/sito o telefono: 091-238.62603 **VASSALLO GIORGIO** Wednesday 14:00 16:00 Edificio 6 - terzo piano.		

DOCENTE: Prof. ANTONIO GENTILE

PREREQUISITES	
LEARNING OUTCOMES	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. 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. 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. 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.
ASSESSMENT METHODS	Prova scritta. Prova orale.
TEACHING METHODS	Lezioni frontali Esercitazioni in aula e nelle aule informatiche

MODULE
MODULE 1

Prof. ANTONIO GENTILE

SUGGESTED BIBLIOGRAPHY

1. M. Morris Mano, Charles R. Kime, “Reti Logiche”, Ed. Italiana, Pearson Education
2. Y. Patt, S. Patel, "Introduction to Computing Systems: From bits & gates to C & beyond", 2nd Ed., McGraw-Hill
3. D. A. Patterson, J. L. Hennessy, “Computer Organization and Design”, Morgan Kaufmann

AMBIT	50283-Matematica, informatica e statistica
INDIVIDUAL STUDY (Hrs)	192
COURSE ACTIVITY (Hrs)	108

EDUCATIONAL OBJECTIVES OF THE MODULE

Al termine del corso lo studente conoscerà i concetti necessari alla comprensione della struttura dei calcolatori elettronici e la loro evoluzione storica. Concetti di base necessari alla comprensione della struttura dei calcolatori elettronici digitali programmabili. Conoscera le principali nozioni sull'algebra di Boole e sulle reti logiche. Avra conoscenza delle problematiche inerenti le metodologie di progettazione di reti logiche combinatorie e sequenziali. Conoscerà la struttura dei calcolatori e dei loro componenti secondo i principali modelli architetturali. Avra conoscenza di base di sistemi operativi Unix-like. Lo studente sara in grado di valutare, analizzare, comunicare e implementare le possibili soluzioni software a problemi applicativi di media complessita utilizzando l’acquisita padronanza del linguaggio C.

SYLLABUS

Hrs	Frontal teaching
6	Calcolatori elettronici digitali programmabili e loro evoluzione storica. Struttura di un calcolatore. Modello Von Neumann: CPU, memoria, input/output. Microprocessori. Discussione
6	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.
6	Rappresentazione delle informazioni. Rappresentazione di interi con segno. Rappresentazione in complemento alla base. Rappresentazione di numeri reali: virgola fissa, virgola mobile. Rappresentazione di caratteri alfanumerici. Codice ASCII. Rappresentazione di immagini (cenni). Discussione.
6	Algebra Booleana Operatori e porte logiche. Funzioni. Tabelle di verita. Diagrammi e circuiti logici. Identita fondamentali. Principio di dualita. Teorema di de Morgan. Complemento di una funzione. Forme canoniche. Mintermini. Maxtermini. Discussione
6	Sintesi a due livelli. Mappe di Karnaugh. Implicanti, primi implicanti e primi implicanti essenziali di funzioni booleane. Minimizzazione di funzioni booleane. Operatore XOR. Operatori funzionalmente completi. Discussione.
6	Reti combinatorie Decoder ed encoder. Espansione in serie di decoder. Encoder con priorita. Multiplexer e demultiplexer. Sintesi con decoder. Sintesi con multiplexer. Sommatori. Dicussione.
6	Reti sequenziali Modelli di Mealy e Moore. Latch. Flip-Flop. Analisi di reti sequenziali sincrone. Sintesi di reti sequenziali sincrone. Discussione
6	Memorie e dispositivi logici programmabili Generalità sui dispositivi di memoria e sui dispositivi logici programmabili. Tecnologie logiche programmabili. Memorie a sola lettura (ROM). Implementazione di circuiti combinatori mediante ROM. Dispositivi a matrice logica programmabile (PAL). Dispositivi logici a matrice programmabile (PLA). Cenni sui dispositivi logici programmabili VLSI e FPGA. Discussione.
6	Introduzione al VHDL. Schematic entry vs. VHDL. Rappresentazione strutturale. Diagrammi di temporizzazione e simulazione. Rappresentazioni in VHDL. Descrizione strutturale. Descrizione dataflow. Descrizioni gerarchiche. Descrizione funzionale. Descrizione in VHDL di circuiti sequenziali. Discussione.
6	Modellazione in VHDL di strutture hardware. Implementazione di reti combinatorie: multiplexer, demultiplexer, decoder, encoder, sommatori e moltiplicatori. Implementazione di reti sequenziali: modelli mealy e moore, macchine a stati finiti, registri e contatori, dispositivi di memoria. Discussione
6	Il calcolatore elettronico: l'architettura di Von Neumann. Componenti di base: memoria, unità di processo, unità di ingresso e uscita, unità di controllo. L'architettura LC-3. Processo di esecuzione delle istruzioni. Modifica dell'ordine delle istruzioni e arresto del flusso. Discussione
6	Architettura del LC-3. l'Instruction Set Architetcure. Istruzioni calcolo e di movimento dati. Istruzioni di controllo. Rivisitazione del data path. Discussione
6	Introduzione alla programmazione. Il problem solving: decomposizione sistematica. I costrutti di base: sequenze, condizioni e iterazioni. I tre costrutti nell'LC-3. Debugging. Discussione.

6	Il linguaggio assemblativo. Scriver un programma in linguaggio assemblativo. Processo di assemblaggio in due fasi: tabella dei simboli, generazione del programma in linguaggio macchina. Combinare più file oggetto. Esempi e discussione.
6	Il sottosistema di ingresso/uscita. Generalità. Ingresso da tastiera, registri di ingresso e routine di servizio. Ingresso mappato in memoria. Uscita su monitor, registri di uscita e routine di servizio. Uscita mappata in memoria. Esempi e discussione
6	Il sottosistema di ingresso/uscita. Ingresso/uscita con interrupt. Ingresso/uscita mappato in memoria. Esempi e discussione.
6	Meccanismi di TRAP e subroutines nel LC-3
6	Lo stack. Struttura di base. Ingresso/uscita con interrupt. Aritmetica attraverso lo stack e conversione dei dati. Esempi e discussione.
6	Implementazione del LC-3 in VHDL. Descrizione funzionale.
6	Implementazione del LC-3 su dispositivi programmabili.

MODULE
MODULE 2

Prof. GIORGIO VASSALLO

SUGGESTED BIBLIOGRAPHY

Al Kelley, Ira Pohl, “C didattica e programmazione”, Pearson

B. W. Kernighan, D. M. Ritchie, "Linguaggio C. II ed.", Gruppo Editoriale Jackson,

AMBIT	50283-Matematica, informatica e statistica
INDIVIDUAL STUDY (Hrs)	96
COURSE ACTIVITY (Hrs)	54

EDUCATIONAL OBJECTIVES OF THE MODULE

RISULTATI DI APPRENDIMENTO ATTESI
Conoscenza e capacità di comprensione
Lo studente alla fine del corso acquisirà una buona conoscenza della programmazione in linguaggio C in un ambiente Unix-like. Sarà in grado di valutare, analizzare e sintetizzare le possibili soluzioni software a semplici problemi. Avrà conoscenza delle problematiche inerenti le metodologie di progettazione di reti logiche combinatorie e le metodologie di progettazione di reti sequenziali. Inoltre avrà una conoscenza di base dell’architettura del calcolatore. Sarà in grado di utilizzare un SO Unix-like.

Capacità di applicare conoscenza e comprensione
Lo studente sarà in grado di utilizzare strumenti e ambienti di sviluppo per la programmazione C in un ambiente Unix-like e di implementare semplici programmi. Sarà in grado di progettare a livello logico funzionale semplici circuiti logici per la soluzione di problemi elementari. Sarà in grado di progettare semplici strumenti software.

Autonomia di giudizio
Lo studente sarà in grado sia di effettuare l'analisi di un problema che di progettare, a partire da una descrizione verbale, una opportuna soluzione software. Sarà in grado di valutarne la qualità di una soluzione software in termini di semplicità, leggibilità, efficienza e possibilità di riutilizzo. Sarà in grado di capire i principi di base della programmazione
.
Abilità comunicative
Lo studente acquisirà la capacità di comunicare ed esprimere problematiche inerenti l’oggetto del corso. Sarà in grado di sostenere conversazioni su tematiche relative alla progettazione di strumenti software. Sarà in grado di utilizzare un linguaggio semplice e chiaro per la descrizione dei processi di analisi e di sintesi di soluzioni software a problemi elementari.

Capacità d’apprendimento
Lo studente dovrà sviluppare la capacità di apprendere i processi di analisi e di sintesi relativi alla codifica di programmi di complessità medio-bassa.

SYLLABUS

Hrs	Frontal teaching
3	Presentazione del modulo. Introduzione alla programmazione. Linguaggi interpretati e compilati. Un esempio di linguaggio interpretato, la bourne shell e la linea dei comandi. Introduzione all'uso della linea dei comandi, panoramica dei principali comandi di shell.
3	programmazione della shell (bash),variabili della shell, redirezione dell'input-output, semplici esempi di script.
2	Introduzione alla programmazione in linguaggio C. Precompilatore del linguaggio C, struttura di un programma C. Tipi e variabili.
2	Compilazione di un programma con il compilatore gcc, variabili intere e in virgola mobile, istruzioni condizionali.
3	Libreria standard del linguaggio C. Istruzioni cicliche. Le funzioni in linguaggio C, parametri - variabili globali e locali.
3	Introduzione all'uso dei puntatori, passaggio dei parametri per valore e per riferimento. Puntatori. Introduzione all'uso dei vettori e delle matrici in linguaggio C.
2	Passaggio di vettori alle funzioni in linguaggio C. Funzioni per il calcolo del valore minimo massimo e medio degli elementi di un vettore.
2	Le stringhe e i vettori di caratteri in linguaggio C. Funzioni di libreria per la gestione delle stringhe. Esempi di utilizzo delle stringhe in linguaggio C.
3	Puntatori e array, aritmetica dei puntatori, funzioni per la gestione delle stringhe in linguaggio C.
2	Funzioni in linguaggio C per la conversione di una stringa rappresentante un numero in una base generica in una variabile intera e viceversa.
2	Operatori bitwise e gestione di insiemi. Conteggio rapido del numero di bit posti ad uno in una variabile intera.
3	Variabili globali. Algoritmo per la generazione di una permutazione casuale. Algoritmo bubble sort su vettori generici. Passaggio di una funzione come parametro. Complessità computazionale del bubble sort. Bubble sort con flag.

3	Ricorsione e algoritmi ricorsivi. Fattoriale, Torre di Hanoi e ricerca dicotomica in un vettore ordinato. Valutazione della complessità computazionale degli algoritmi.
3	Introduzione alle strutture. Gestione di numeri complessi in linguaggio C. Creazione di una libreria per la gestione di numeri complessi in forma algebrica ed euleriana.
3	Introduzione alla gestione dei file di testo e dei file binari in linguaggio C. Funzioni della libreria standard per la gestione dei file.
3	Introduzione alle strutture dati. Esempio di struttura dati dinamica: le liste. Funzioni per la gestione di liste concatenate. Implementazione di uno stack. Funzioni principali dello stack.
2	Gestione di immagini con il formato pgm e ppm in linguaggio C.

Hrs	**Practice**
4	Implementazione di semplici script di shell. Utilizzo dei principali comandi della shell. Script per la gestione e l'archiviazione di directory e per l'elaborazione non interattiva di file di testo.
4	Implementazione di semplici programmi in linguaggio C. Esercizi sulla gestione delle stringhe. Funzioni per la gestione di vettori e matrici.
4	Implementazione in linguaggio C di semplici operazioni su vettori, stringhe e matrici. Moltiplicazione riga per colonna di matrici. Gestione di insiemi con operatori bitwise. Implementazione di programmi che utilizzano i numeri complessi.
4	Implementazione di un programma per la simulazione a livello logico funzionale di una rete sequenziale. Implementazione di una semplice calcolatrice interattiva in linguaggio C.