UNIVERSITÀ DEGLI STUDI DI PALERMO

DEPARTMENT	Ingegneria		
ACADEMIC YEAR	2020/2021		
BACHELOR'S DEGREE (BSC)	CYBERNETIC ENGINEERING		
INTEGRATED COURSE	ELECTRONIC CALCULATORS - INTEGRATED COURSE		
CODE	18794		
MODULES	Yes		
NUMBER OF MODULES	2		
SCIENTIFIC SECTOR(S)	ING-INF/05		
HEAD PROFESSOR(S)	LA CASCIA MARCO	Professore Ordinario	Univ. di PALERMO
OTHER PROFESSOR(S)	CONCONE FEDERICO	Ricercatore a tempo determinato	Univ. di PALERMO
	LA CASCIA MARCO	Professore Ordinario	Univ. di PALERMO
CREDITS	12		
PROPAEDEUTICAL SUBJECTS			
MUTUALIZATION			
YEAR	1		
TERM (SEMESTER)	Annual		
ATTENDANCE	Not mandatory		
EVALUATION	Out of 30		
TEACHER OFFICE HOURS	**CONCONE FEDERICO** Friday 17:00 18:00 Laboratorio di Intelligenza Artificiale e Sistemi Distribuiti, Edificio 6, Terzo piano **LA CASCIA MARCO** Monday 15:00 17:00 Microsoft Teams Codice: wztkv0u		

DOCENTE: Prof. MARCO LA CASCIA

PREREQUISITES	Since the course is in the first year and its duration is one year, the contents of the course are selfcontaining and do not require prerequisites
LEARNING OUTCOMES	AWAITED RESULTS OF THE LEARNING Knowledge and comprehension capabilities At the end of the course, the student will have a good knowledge of the techniques of information representation and its processing through combinatorial and sequential circuits. He/she will know the technique for the optimization of these systems. He/she will know the syntax and the technique for the programming with C language. He/She will be able to employ data stuctures and the fundamental algorithms. Comprehension and capability to apply knowledge At the end of the course, the student will be able to apply the studied techniques to analyse and design, at a functional logic level, combinatorial and sequential nets. The student will be able to use tools and environments for the programming with the C language and implement software solutions. Autonomy of Judgement The student will be able to analyse a problem and design, starting from a verbalil description, a suitable software solution. He/she will have the capability to evaluate the quality of a software solution according efficiency, readability, semplicity and reusability. He/she will be able to understand the working principles of computers. Communication capabilities The student will acquire the capability to understand and express problems involving the contents of the course. He/she will be able to describe with proper terms a logic circuit and, using with a clear language, to describe the processes of writing and analysis of software solutions. Learning capabilities The student will acquire capabilities for the application of the studied methodologies in different contexts and to learn processes of analysis and design of both software solutions and logic circuits.
ASSESSMENT METHODS	The learning process is evaluated through two tests one for each module of the course. In the test dealing with the digital circuits module the student is asked to answer close-ended questions, open-ended questions and design of sequential and combinatorial circuits. The minimal number of questions is five and the exam lasts two hours. The tests are aimed at verifying : a) acquired knowledge b) the capability in designing circuits according given requirements c) the capability to organize and present the technical knowledge. The maximum score is assigned if are satisifed three aspects: the capability to answer questions about the representation of the information, the capability to design and optimize logical circuits that must meet the given requirements, describe and compare different circuital solutions. The test dealing with fundamentals of computer science module is held in the computer room. The students have to prepare programs, written in C language, according the requirements of the questions. The minimum number of question is four and the time for the trial is two hours. The questions of the test are aimed at verifying a) the knowledge of the programming language; b) the capability to find simple solution to typical problems; c) the capability to create working programs The maximum score is obtained if it is verified the full competence in three aspects: the use of correct syntactic forms of the programming language, the capability to compose language structures to solve problem dealing with the acquisition, processing and storage of information; the creation of programs that can be executed without presenting errors or failures. GRADES 30-30 and laude: Excellent. Full knowledge and understanding of concepts and methods of the discipline, excellent analytical skills even in solving original problems; excellent communication and learning skills. 27-29: Very good. Very good knowledge and understanding of concepts and methods of the discipline; very good communication skills; very good capability of concepts and methods applications. 24-26: Good. Good knowledge of main concepts and methods of the discipline; discrete communication skills; limited autonomy for applying concepts and methods for solving original problems. 21-23: Satisfying. Partial knowledge of main concepts and methods of the

	discipline; satisfying communication skills; scarce judgment autonomy. 18-20: Acceptable: Minimal knowledge of concepts and methods of the discipline; minimal communication skills; very poor or null judgement autonomy. Non acceptable: Insufficient knowledge and understanding of concepts and methods of the discipline.
TEACHING METHODS	The couse is composed by lessons and by exercises in classrom. During lesson concepts and notions are exposed to the class. The exercises are used to provide example of application of the studied materials and to self assess the acquired knowledge

MODULE
PRINCIPLES OF COMPUTER SCIENCE

Prof. FEDERICO CONCONE

SUGGESTED BIBLIOGRAPHY

Paul J. Deitel, Harvey M. Deitel, Il linguaggio C, Pearson Kernighan
Brian W., Ritchie Dennis M., Il linguaggio C Principi di
programmazione e manuale di riferimento, Pearson Education
Italia Al Kelley, Ira Pohl, C didattica e programmazione, Pearson

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

EDUCATIONAL OBJECTIVES OF THE MODULE

At the end of the course, the student will be able to evaluate, analyze and implement the possible software solutions for simple problems by using C programming language.

SYLLABUS

Hrs	Frontal teaching
6	Introduction to the course. Programming basic concepts. The alghoritms. The Flow chart and the basic blocks.
5	Source program and object program. The compilers and the interpreters. The interpreted language and the compiled language. The procedural language and the object language programming. Short information about the linker. Introduction to language C programming. The case-sensitive problem. C pre-compiler.
6	Standard library of C language. Types and variables. Simple and structured variables. I/O of numerical values. Boolean and mathematical statements, operators' priority. The characters. The pointers.
8	Selection structures, nested structures and alternative evaluations. Differences between if and the switch-case constructs. States programming. The iterative structures while, do-while e for. Vectors and matrices. Typical algorithms involving matrices, the Bubble sort, Quick sort, Merge sort algorithms.
6	The string type and the characters' vectors. Functions libraries for managing strings. Functions for the representation of an integer based on a generic numeration system. Functions for the conversion of a string representing an integer.
7	Sequential and random access files. Management of text and binary files in C language. Functions for managing text and binary files: fopen, fclose, fscanf, fprintf, fread, fwrite, fseek
6	Functions, prototypes and definition of functions. Function calls. Parameters. Call functions by parameter values or by reference. Recursion (factorial calculation, the tower of Hanoi).
10	The structures. Simple examples of data structures in C language. Linked lists. Insertion in head and tail. Deleting a list item. Other implementations with structures (graphs, binary trees). Brief Introduction to Linux OS

MODULE
LOGIC CIRCUITS

Prof. MARCO LA CASCIA

SUGGESTED BIBLIOGRAPHY	
- M. Morris Mano, Charles R. Kime, Reti Logiche, Edizione italiana, Pearson Education Italia, ISBN: 88-7192-142-9	
AMBIT	50283-Matematica, informatica e statistica
INDIVIDUAL STUDY (Hrs)	96
COURSE ACTIVITY (Hrs)	54
EDUCATIONAL OBJECTIVES OF THE MODULE	
The student will be able to apply in different contexts the methodologies learnt during the course, and to learn processes of analysis and design of logic networks and circuits.	

SYLLABUS

Hrs	Frontal teaching
12	Introduction, basic concepts of information and its logarithmic nature. How to represent numbers in various bases, including binary numbers. How to convert number from one base to another. How to represent floating point numbers and negative numbers. How to represent images and text, including ASCII code. Introduction to Boolean algebra, logic gates, boolean functions, theorems and properties of Boolean algebra, normal forms, Karnaugh maps, minimization of Boolean function. Design of combinational circuits. Decoders, Multiplexers, Full adder. Design of combinational networks
16	Sequential networks. Mealy and Moore models. Latches (SR, D), Flip-Flops (JK, T, D, SR). Master slave. Analysis and design of synchronous sequential circuits. State diagrams, Flip Flop characteristic equations, Design of sequential networks with Flip Flops of various types (D, T, JK, SR). Sequence recognizers. Registers, counters, ROM, PAL, PLA
8	Computer architecture (von Neumann model), CPU, ALU, RAM, BUS. What is a program. CISC and RISC systems. Microprograms and wired logic. Introduction to operating systems

Hrs	Practice
4	How to represent information. Boolean algebra
7	Analysis and Design of combinational networks. Karnaugh maps; Normal forms
7	Analysis and Design of sequential networks. Design of sequence recognizers