UNIVERSITÀ DEGLI STUDI DI PALERMO

DEPARTMENT	Ingegneria		
ACADEMIC YEAR	2022/2023		
BACHELOR'S DEGREE (BSC)	DIGITAL ENTERPRISE INNOVATION ENGINEERING		
INTEGRATED COURSE	ELECTRONIC CALCULATORS - INTEGRATED COURSE		
CODE	18794		
MODULES	Yes		
NUMBER OF MODULES	2		
SCIENTIFIC SECTOR(S)	ING-INF/05		
HEAD PROFESSOR(S)	GAMBINO ORAZIO	Ricercatore	Univ. di PALERMO
OTHER PROFESSOR(S)	GAMBINO ORAZIO	Ricercatore	Univ. di PALERMO
CREDITS	12		
PROPAEDEUTICAL SUBJECTS			
MUTUALIZATION			
YEAR	1		
TERM (SEMESTER)	Annual		
ATTENDANCE	Not mandatory		
EVALUATION	Out of 30		
TEACHER OFFICE HOURS	**GAMBINO ORAZIO** Monday 10:00 12:00 Chat di Teams, previo appuntamento concordato via email. Tuesday 10:00 12:00 Chat di Teams, previo appuntamento concordato via email.		

DOCENTE: Prof. ORAZIO GAMBINO

PREREQUISITES	Basic use of the computer with Windows operating system: copy, paste and rename files, installation of applications, use of the graphic interface.
LEARNING OUTCOMES	Knowledge and understanding The student will acquire in-depth knowledge of structured programming in C language. He will know the main programming tools. It will acquire elements of information representation in computers and basic methodologies for the design and analysis of combinatorial and sequential logic networks. The student will acquire basic knowledge on computer architectures. Ability to apply knowledge and understanding The student will be able to evaluate possible software solutions to problems of medium complexity and address their implementation using development tools and environments for programming in C language. He will be able to tackle simple problems of binary representation of information. Will be able to functionally design logic circuits for solving simple problems. Autonomy of judgment The student will be able to independently deal with the analysis, design and implementation of software using structured programming. He will be able to evaluate the quality of the software in terms of simplicity, readability, structure and efficiency. Communication skills The student will be able to expose, effectively and with language properties, analysis and solutions of problems that can be faced with structured programming and functional design of logic circuits, as well as information representation problems. Learning skills The student will be able to independently tackle structured programming problems by identifying and integrating partial solutions already available, both formalized and implemented. He will be able to autonomously deepen his knowledge of software modules and programming interfaces. Will be able to deepen the knowledge of the languages and paradigms of programming, operating systems, computer architectures and logic circuits.
ASSESSMENT METHODS	The assessment of learning takes place through an ongoing test for the Logic Networks module and the development of code to be written in C language according to a theme assigned by the teacher for the Programming Fundamentals module. The in itinere test consists in the compilation of a multiple choice test concerning the part of logic networks and assembly programming. The questionnaire includes fifteen multiple-choice questions on topics related to logic networks, the structure of electronic computers and the representation of low-level information. The answer to each question is evaluated with a score of 2, if the answer is correct, 0 if the answer is not provided, -1 if the answer is incorrect. The test grade, out of thirty, is obtained by adding the scores of all the answers to the questions. The ongoing test is deemed to have been passed with a grade equal to or greater than 18/30. For the test of the Programming Fundamentals module, the source code in C language must be written according to a text assigned by the teacher. The code must be able to be compiled and the executable must be functional. The code is evaluated by the teacher according to the following characteristics: modularity of the code, indentation, presentation of the result, adaptability to data of the same nature and adherence to the proposed problem. Students not in progress or who have not passed the in itinere test, are required to take an overall test integrating the two modules, with the same modalities, respectively, of the in itinere test and the final computer test. The grades of the ongoing test on the logic network module and of the Fundamentals of programming are added together with weights respectively of 0.3 and 0.7 (for a score of Fundamentals of Programming> 18/30) to obtain an overall evaluation out of thirty; otherwise, the vote of Fundamentals of Programming will be valid for the whole subject. The final mark out of thirty, in the interval 18 / 30-30 / 30 with honors, is obtained

	by averaging the evaluations of the Logic Networks and the Fundamentals of Programming tests. Honors are awarded to students who have obtained 30/30 for both tests and who correctly answer an oral question. The formulation of the tests provides an evaluation of the expected results in relation to the final grade as follows: - from 18/30 to 20/30: sufficient knowledge and understanding of the topics covered, ability to apply the knowledge acquired for the resolution of the proposed problems, autonomy of judgment, communication skills and ability to learn. - from 21/30 to 23/30: fair knowledge and understanding of the topics covered, ability to apply the knowledge acquired for the resolution of the proposed problems, autonomy of judgment, communication skills and ability to learn. - from 24/30 to 26/30: good knowledge and understanding of the topics covered, ability to apply the knowledge acquired for the resolution of the proposed problems, independent judgment, communication skills and ability to learn. - from 27/30 to 30/30 cum laude: excellent knowledge and understanding of the topics covered, ability to apply the acquired knowledge for the resolution of the proposed problems, autonomy of judgment, communication skills and ability to learn. The formulation of the tests provides an evaluation of the expected results in relation to the final grade as follows: - from 18/30 to 20/30: just sufficient knowledge and understanding of the topics covered, ability to apply the acquired knowledge for the resolution of the proposed problems, autonomy of judgment, communication skills and ability to learn. - from 21/30 to 23/30: fair knowledge and understanding of the topics covered, ability to apply the knowledge acquired for the resolution of the proposed problems, autonomy of judgment, communication skills and ability to learn. - from 24/30 to 26/30: good knowledge and understanding of the topics covered, ability to apply the acquired knowledge to solve the proposed problems, independent judgment, communication skills and ability to learn. - from 27/30 to 30/30 cum laude: excellent knowledge and understanding of the topics covered, ability to apply the acquired knowledge for the resolution of the proposed problems, autonomy of judgment, communication skills and ability to learn.
TEACHING METHODS	The lessons will be held by the teacher in the classroom with video projection support which will show the teacher's computer desktop. The students can bring their notebooks into the classroom to test the code shown on the projection screen.

MODULE
LOGIC CIRCUITS

Prof. ORAZIO GAMBINO

SUGGESTED BIBLIOGRAPHY

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

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

EDUCATIONAL OBJECTIVES OF THE MODULE

The module aims to provide students with the basic concepts regarding: design of logic circuits and sequential networks, computer architecture, structure and operation of microprocessor, Intel 8086 Assembly programming.

SYLLABUS

Hrs	Frontal teaching
4	Introduction to the course. Exam modality. Internal and external representation. The analog signal vs digital signal.
6	Conversion from base 10 to base 2. Conversion of fractional numbers. Additions and subtractions in base 2. Numeric codes: BCD, ASCII, Gray. IEEE 754 standard. Representation of images and sounds.
3	Boolean Algebra. Operators and logic gates. Functions. Tables of truth. Diagrams and logic circuits. Fundamental identities. Duality principle. De Morgan's theorem. The complement of a function. Canonical forms. Minterms. Maxterms. Synthesis on two levels.
6	Maps of Karnaugh. Implicants, prime implicants and essential prime implicants principles of Boolean functions. Minimization of Boolean functions. XOR operator. Functionally complete operators.
4	Combinatorial networks. Decoder and encoder. Expansion of decoders in series. Encoder with priority. Multiplexer and demultiplexer.
3	Combinatorial networks. Synthesis with decoder. Synthesis with multiplexer. Half-adder and Full-adder. ALU 74181
4	Sequential networks. Latch. Flip-Flop.
3	Sequential networks: counters and registers.
5	von Neumann Architecture. CPU and RAM. ROM memory. Mass storage. Cache.
4	Intel 8086 microprocessor structure. Description and operation of BIU and EU units. 8086 microprocessor registers. Memory segmentation.
12	Assembly programming 8086. Structure of an assembly program. Memory models. Addressing methods. Offset. Vettori. Interrupt. LEA. Subroutines. 8 and 16 bit addition. Stack. Multiplication Conditional and unconditional jumps. Cycles. Scrolling and rotation.

MODULE
PRINCIPLES OF PROGRAMMING

Prof. ORAZIO GAMBINO

SUGGESTED BIBLIOGRAPHY

- J. Glenn Brookshear, Stephen G. Kochan, “Fondamenti di informatica e programmazione in C”, Pearson
- B. W. Kernighan, D. M. Ritchie, “Linguaggio C”, Pearson
- Slides proiettate durante la lezione che sono aggiornate di anno in anno.

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

EDUCATIONAL OBJECTIVES OF THE MODULE

The module aims to provide students with the basic concepts of structured programming in C language. Students will be able to analyze, communicate and implement possible software solutions to application problems using the acquired command of C language.

SYLLABUS

Hrs	Frontal teaching
2	Programmation module presentation. Description of the C program "Hello World!". Analog signal vs digital signal
2	Printf. Data type. Visualization string format for printf. How to comment the code. Conversion from a data type to another: intrinsic and casting.
3	Pre- / Post- increment of a variable. Syntax of the for loop. Sum from 1 to 13 through cycles both in asm8086 and C. For cycle for countdown. Partial sum algorithm for Gauss. Meaning of pseudocode. Definition of heuristics and algorithm. Nested for loops. Relational operators.
3	While cycle with advance evaluation. Postponed evaluation do..while cycle. Nested while w do ... while loops. Flowcharts. Danger of the infinite loop. Bohm and Jacopini's theorem. The selection: the if ... else construct. Switch...case construct .
3	Boolean operators for expressions. Boolean type. Bitwise Boolean operators. Introduction to one-dimensional arrays. Accented and special characters of the printf. Preprocessor macro. Ternary operator vs digital signal
3	One-dimensional array. Array initialization by enumeration and by loop. Array with / without size. Examples of using the array. Const qualifier
3	Dot product between vectors with comparison with Assembly 8086. Two-dimensional arrays. Linearization of matrices also with macros. Row major and column major. Good and bad programming practices with arrays. Copy from vector to matrix.
3	Swap between variables. Transposed matrix into a new result matrix and on the same matrix. Elimination Gauss-Jordan. Matrix product. Introduction to Functions in C.
3	Local variables. Examples with local variables. Formal and current parameters. Advantages of using functions and difference with macros with parameters. Values returned by the function. Passing of variable to one by value (per copy).
3	Functions that call functions. Variable passing by address. Pass by address of arrays by default. Minimum / maximum of a vector. Sorting by selection of a vector. Function prototype.
2	Global variables. Recursive algorithms.
3	Struct. Global Struct. Passing a default struct. Functions that call functions containing structs.
2	Array Struct. Array of struct. Introduction to character strings.
3	Merge and count words in character strings. Keyboard input with getchar. Conversion from string to integer.
3	String.h library. Complexity order of algorithms: linear case. Sequential search.
3	Complexity order of algorithms: the logarithmic case for binary search, the quadratic case with sorting by selection.
2	File management. Stdin-Stdout-Stderr. Introduction to pointers.
3	Values swap and addressing between variables by means of pointers. Passing a variable to a function by address. Passing a struct by address using pointers.
2	Dynamic array
3	Lists