



UNIVERSITÀ DEGLI STUDI DI PALERMO

DEPARTMENT	Matematica e Informatica		
ACADEMIC YEAR	2019/2020		
BACHELOR'S DEGREE (BSC)	MATHEMATICS		
INTEGRATED COURSE	PROGRAMMING WITH LABORATORY		
CODE	10664		
MODULES	Yes		
NUMBER OF MODULES	2		
SCIENTIFIC SECTOR(S)	INF/01		
HEAD PROFESSOR(S)	TEGOLO DOMENICO	Professore Associato	Univ. di PALERMO
OTHER PROFESSOR(S)	TEGOLO DOMENICO	Professore Associato	Univ. di PALERMO
	EPIFANIO CHIARA	Ricercatore	Univ. di PALERMO
CREDITS	9		
PROPAEDEUTICAL SUBJECTS			
MUTUALIZATION			
YEAR	1		
TERM (SEMESTER)	1° semester		
ATTENDANCE	Not mandatory		
EVALUATION	Out of 30		
TEACHER OFFICE HOURS	EPIFANIO CHIARA Tuesday 14:30 17:00 Dipartimento di Matematica ed Informatica, via Archirafi 34, Room 104, primo piano/ first floor		
	TEGOLO DOMENICO Wednesday 15:00 17:00 Dipartimento Matematica e Informatica Via Archirafi 3490123Palermo		

PREREQUISITES	No prerequisite is required
LEARNING OUTCOMES	<p>Knowledge and understanding: Acquisition of the fundamentals of computer systems and of the principles of structured programming; Learning the basic concepts on static and dynamic data structures; Understanding simple fundamental algorithms on alphanumeric sorting or research; Assimilate the concepts on the solution of simple problems through recursive programming; possession of the competencies on the fundamental constructs of the programming language C.</p> <p>Ability to apply knowledge and understanding: Understanding the cognitive processes useful for identifying a solution to a simple or complex numerical problem. Programming capacity in the programming language C. Ability to identify and resolve syntactic and semantic errors that will emerge during the compilation phase, and the execution. Be able to break down complex problems into elementary problems.</p> <p>The autonomy of judgment: Be able to evaluate the goodness of methods and content in the preparation of a program. Broad understanding of advanced concepts on the use of constructs and data structures. Be able to evaluate the implications on the use of constructs and on passing parameters to functions. The achievement of independent judgment on the applications developed based on an efficient algorithmic solution.</p> <p>Communication skills: Properties of expression in the presentation of the basics of the art of programming and the programming language C.</p> <p>Learning capacity: Know how to approach programming, problem-solving and knowing how to contextualize the skills acquired in concrete problems.</p>
ASSESSMENT METHODS	<p>The final grade will take into account the grade of the middle term exams, moreover the final exam including an oral interview or alternatively an account of the grade obtained in the final integrated test consisting of a written test on the topics of the two modules and an oral examination.</p> <p>The written tests will consist of 10 between multiple-choice or open questions, and 2 programming exercises. The questions will be evaluated with a score from 0 to 2, while each exercise will have a maximum score of 5, the written test will be considered passed with a score greater than or equal to 15/30.</p> <p>The oral exam, assessed in thirtieths, will consist of the discussion of the written tests (middle tests or final written test), and of questions in order to validate the knowledge of the course topics.</p> <p>The final grade will take into account the grade reported in the middle tests or in the final written and oral exam.</p> <p>Therefore, it is reiterated that some written tests are scheduled according to the didactic calendar whose nature is very similar to the written tests of the exam sessions. partial or total) of the final written test.</p> <p>The final evaluation will be based on the following conditions: a) Very good, wide and high knowledge of the proposed topics; ability to apply them with rigour and in full autonomy; possession of excellent communication skills (grade 29-30L). b) Good knowledge of the proposed topics and ability to apply them with mathematical and methodological rigour, but not in full autonomy; possession of good language properties (grade 26-28); c) Discrete knowledge of the proposed topics and sufficient ability to apply them independently; good ability to complete a rigorous reasoning and good language properties (grade 22-25); d) Basic knowledge of the proposed topics and limited ability to apply them independently; sufficient ability to complete a rigorous reasoning and sufficient language ownership (grade 18-21); e) Insufficient knowledge: the student does not possess an acceptable knowledge of the contents and of the topics covered in the teaching and has no ability to autonomously apply the acquired knowledge.</p>
TEACHING METHODS	The achievement of the teaching objectives will be achieved through frontal lessons and laboratory experiences.

MODULE STRUCTURED PROGRAMMING

Prof. DOMENICO TEGOLO

SUGGESTED BIBLIOGRAPHY

Paul J. Deitel - Harvey M. Deitel, Il linguaggio C - Fondamenti e tecniche di programmazione • 8/Ed.
A. Bellini, A.Guidi. Linguaggio C - guida alla programmazione. Mc Graw Hill.

AMBIT	50194-Formazione informatica
INDIVIDUAL STUDY (Hrs)	78
COURSE ACTIVITY (Hrs)	72

EDUCATIONAL OBJECTIVES OF THE MODULE

The module aims to provide theoretical and workroom methodologies aimed at acquiring the basic concepts for programming in computer system environment. Both the representation of data and the elementary constructs of an imperative programming language will be analyzed, and simple algorithms will be used, which will use control, sequencing, selection and iteration structures.

Due to propaedeutic and considerable diffusion in the market, and in order to guarantee more versatility in the world of work, the programming language C will be taken into account.

SYLLABUS

Hrs	Frontal teaching
2	Introduction to the module, organization of a computer, hardware evolution, evolution of operating systems, evolution of programming languages.
4	Computer: Hardware and software. The binary system: definition, operations. Switching from decimal to binary and vice versa. Information and measurements: bit and byte. Introduction to C development environments, introduction to programming. Definition of Algorithm. Simple meta-language programs. Understand the complexity of an algorithm.
4	Introduction to different programming paradigms: imperative paradigm (structured programming and object programming), declarative paradigm (functional programming and logic programming). Structured programming. The Böhm-Jacopini theorem. The C language and the structures of a program. The Sequence construct, If ... else, switch..case constructor. Identifiers. Input / output functions.
2	Constants and variables, instruction of assignment. Standard data types: Integer, character and their representation, float and double types. ASCII code and other character codes. Representing real numbers in memory. Operators in C and their priority. Incremental and decreasing operators of a integer variable.
4	Loop control structures: The loop control FOR, the loop control WHILE, the loop control WHILE ...DO. Equivalence among loop controls.
2	Array in C. Array unidimensional and applications. Definition and display of arrays. N-dimensional arrays: matrices. Definition of assignment and display of a matrix.
2	Linear Search, Iterative Binary Search. Sorting Algorithms. SelectionSort, InsertionSort, BubbleSort, MergeSort, QuickSort. Strings and Function Library on strings <string.h>.
2	Functions in C: Declaring, Defining and Calling Functions. How to send parameters to a function and visibility of variables.
2	Pointers. Array and Pointers. Arithmetic of pointers. Loop functions on arrays. Recursion. Recursive functions and algorithms.
Hrs	Workshops
4	Laboratory with exercise on assignment, increase and decrease operators. Basic Input / Output Instructions
4	Laboratory with arithmetic exercise on Integer and Real variables. Equal and Relational Operators in C.
4	Laboratory with exercise on selection constructions with and without nesting.
4	Laboratory with exercise on iterative construct FOR
4	Laboratory with exercise on iterative construct WHILE
4	Laboratory with exercise on iterative construct DO..WHILE
4	Laboratory with tutorial on multiple selection construction SWITCH and BREAK-CONTINUE.
4	Laboratory with tutorial on logical operators and the equivalence of the iterative cycles.
4	Laboratory with tutorial about standard and user-defined function: prototypes of a function and parameters passage. Rules of visibility.
4	Laboratory with a tutorial on arrays: search and sorting. Laboratory on: SelectionSort, InsertionSort, BubbleSort, MergeSort, QuickSort.
4	Laboratory and tutorial on pointers, expressions and arithmetic. Relationship between pointer and array.

4	Laboratory and tutorial on recursion programming; an example of an algorithm: the Hanoi Towers. File Management.
<p style="text-align: center;">MODULE ADVANCED PROGRAMMING <i>Prof.ssa CHIARA EPIFANIO</i></p>	
SUGGESTED BIBLIOGRAPHY	
- C. Demetrescu, I. Finocchi, G.F. Italiano, Algoritmi e strutture dati, McGraw-Hill.	
- A. Bellini, A.Guidi. Linguaggio C - guida alla programmazione. Mc Graw Hill.	
AMBIT	10709-Attività formative affini o integrative
INDIVIDUAL STUDY (Hrs)	39
COURSE ACTIVITY (Hrs)	36
EDUCATIONAL OBJECTIVES OF THE MODULE	
This part of the course deepens some advanced topics concerning programming. In particular we analyze some dynamic data structures defined with the aid of pointers.	

SYLLABUS

Hrs	Frontal teaching
3	Unidirectional pointer lists: creating a list, inserting and deleting a cell.
2	Bidirectional pointer lists, Stacks and Queues.
3	Trees. Binary trees and their traversals.

Hrs	Practice
2	Exercises on the creation of a unidirectional list.
2	Exercises on the insertion and deletion of a cell in a unidirectional list.
2	Exercises on the bidirectional lists.
2	Exercises on stacks and queues.
2	Exercises on the creation of a binary tree.
2	Exercises on tree traversals.

Hrs	Workshops
3	Laboratory on the creation of a unidirectional list.
3	Laboratory on the insertion and deletion of a cell in a unidirectional list.
2	Laboratory on the bidirectional lists.
2	Laboratory on stacks and queues
3	Laboratory on the creation of a binary tree.
3	Laboratory on tree traversals.