NATIONAL UNIVERSITY OF ENGINEERING
COLLEGE OF GEOLOGICAL, MINING AND METALLURGICAL ENGINEERING
GEOLOGICAL ENGINEERING PROGRAM

BIC01 – INTRODUCTION TO COMPUTING

I. GENERAL INFORMATION

CODE	**:**	MA401 Informatics
SEMESTER	**:**	4
CREDITS	**:**	3
HOURS PER WEEK	**:**	5 (Theory – Practice)
PREREQUISITES	**:**	MA333 Mathematics III
CONDITION	**:**	Compulsory

II. COURSE DESCRIPTION

The purpose of this course is to train students in the use of programming languages so they can be applied in later courses and professional career. This course deals with subjects such as: algorithms, standard functions, decision, repetition, selection and control instructions, array, functions and pointers. Students develop computer programs applied to diverse engineering problems.

III. COURSE OUTCOMES

1. Understand the steps for solving computer programming problems.
2. Analyze and design algorithms for solving a specific problem.
3. Develop programs using a programming language in the solution of problems of practical cases applied to engineering projects.
4. Understand step by step the developing of computer programs.
5. Properly use programming languages for the scientific and technological research.

IV. LEARNING UNITS

1. PROGRAMMING FUNDAMENTALS / 8 HOURS

Introduction / Definition of problems and casuistries / Concept of program and instructions /Programming languages and source code / programming steps / Concepts and characteristics of algorithms / Decision, repetition and control algorithms / Flow diagrams and their symbols / Diagramming and pseudocodes / Diagrams with decisions and repetitions.

2. INPUT/OUTPUT INSTRUCTIONS AND OPERATIONS / 8 HOURS

Structure of a program, types and methods / Programming rules and its libraries / data, constant and variables concept / Types of data / variable declaration: int, float, char, long, double/ Input instructions Console()/ output instructions Console.Write()/ Arithmetic operators and their rules / mathematical operations, priorities and use of parenthesis / Assignment statement and its rules / Type conversions / Math functions: / Multiple assignments and include operator / Operators: ++.

3. REPETITION AND DECISION INSTRUCTIONS / 8 HOURS

Logic operators: ==, !=, >, <, >=, <= / Logic operators: !, &&. || / Rules / Logics. Algorithms and diagrams with decisions. / Logics / Instruction “if-else”, syntax and rules / Instruction “only If”, “if-else” with bocks, “If within if” / Algorithms and programs with “if-else” / Instruction “while”, syntax and rules / Instructions “while” with an instruction and with a block of instructions / Instructions do-while, syntax and rules / Algorithms and programs with do-while.

4. SELECTION AND CONTROL INSTRUCTIONS / 8 HOURS

Instruction “for”, start expression. Assessment and preparation / Control, syntax and rules / Instruction “for” with one instruction and with a block of instructions / “For” nested instructions, “internal” for and “external” for / Algorithms and programs with “for” and “nested for”, “nesting” / Until instruction “for” / Switch-case instruction, syntax and rules / Break instruction, syntax and rules / Use of “break” within switch-case / “Continue” instructions.

5. ARRAYS / 8 HOURS

Concept of array, declaration, subscript. Syntax and rules / reading and writing of arrays of one dimension / Sorting and deleting of the elements of an array / Operations with one-dimensional arrays / String arrays, string reading and writing / Comparisons and initializations of strings / String internal functions / Two-dimensional arrays, syntax and rules / Subscripts, addresses and declarations of two-dimensional arrays / Reading, writing and operations with two-dimensional arrays / Multidimensional arrays, syntax, rules and uses / Matrix operations.

6. FUNCTIONS / 8 HOURS

Concept of function, syntax and rules / Definition of a function, arguments and parameters / Local and global variables / Call of a function and transfer values / Typical variables of a function, repeated execution of a function / Return instruction / One-function algorithms and programs / Programs with several function and their iterative function / Concept of recursion / programs with function recursion / Functions with scripts / Reading and writing functions.

7. POINTERS / 8 HOURS

Concept of pointer and pointer assignment / Values exchange among arguments and parameters / Declaration, rules, syntax and pointers / Pointers to arrays / Pointers to scripts / Algorithms and programs with pointers / Concept of pointer array, syntax and rules / Pointer array addressing / Algorithms and programs with pointer array / Algorithms and programs with pointer’s pointer.

V. LABORATORY EXPERIENCES

Lab 1: Decision instructions.
Lab 2: repetitive instructions.
Lab 3: Array and selection sentences.
Lab 4: Functions and pointers.

VI. METHODOLOGY

Sessions will be carried out stimulating students’ active participation, through practical cases programming. Students will form groups for researching and exchanging learning and work experiences. The instructor’s expositions will guide every programming work and they will also advised be individually and in groups using real applications. Lab practical sessions will complement knowledge and develop students’ skills and abilities in problem solving through programming language techniques. Papers’ originality and creativity will be motivated encouraging a constant research.

VII. EVALUATION FORMULA

The average grade PF is calculated as follows:

$$PF = 0.25\,EP + 0.50\,EF + 0.15\,PP + 0.10\,PL$$

EP: Mid-Term Exam
EF: Final Exam
PP: Average of five quizzes
PL: Average of four laboratory reports

VIII. BIBLIOGRAPHY

1. **JOYANES, LUIS**
   C# Programming
   Mc Graw – Hill, 2001

2. **SCHILDT, Herbert**
   Turbo C Programming
   Osborne/McGraw-Hill, 2010

IX. COURSE CONTRIBUTIONS TO STUDENT OUTCOMES ATTAINMENT

Course contributions to Student Outcomes are shown in the following table:

Level 1: Know
Level 2: Comprehend, calculate
Level 3: Model, apply, solve
Level 4: Apply at advanced level, design. Achievement of Student Outcome

Outcome	Contribution
1. Engineering Design Design and integrate systems and components for the discovery and development of subsurface earth resources, and construction of earthworks, satisfying requirements, and given technical, economic, social and legal constraints.	2
2. Problem solving Identify, formulate and solve engineering problems properly using the methods, techniques and tools of geological engineering.	2
3. Sciences Application Apply the knowledge and skills of mathematics, sciences and engineering to solve geological engineering problems.	2
4. Experimentation and Testing Conceive and conduct experiments and tests, analyze data and interpret results.	2
5. Modern Engineering Practice Use and apply techniques, methods and tools of modern engineering necessary for the practice of geological engineering.	2
6. Engineering Impact Understand the impact of geological engineering solutions on people and society in local and global contexts.	2
7. Project Management Determine the budgets, schedules and feasibility of engineering projects, and participate in its management for the attainment of goals.	
8. Environmental Appraisal Take into account the importance of preserving and improving the environment in the development of their personal and professional activities.	
9. Lifelong Learning Recognize the need to keep their knowledge and skills up-to-date according to advances of geological engineering and engage in lifelong learning.	2
10. Contemporary Issues Know and analyze relevant contemporary issues in local, national and global contexts.	2
11. Ethics and Professional Responsibility Evaluate their decisions and actions from a moral perspective and assume responsibility for the executed projects.	2
12. Communication Communicate clearly and effectively in oral, written and graphical formats, interacting with different types of audiences.	2
13. Teamworking Appraise the importance of teamworking and participate actively and effectively in multidisciplinary teams.	2