NATIONAL UNIVERSITY OF ENGINEERING

COLLEGE OF GEOLOGICAL, MINING AND METALLURGICAL ENGINEERING

GEOLOGICAL ENGINEERING PROGRAM

BIC01 – INTRODUCTION TO COMPUTING

I. GENERAL INFORMATION

CODE	**:**	BIC01 Introduction to Computing
SEMESTER	**:**	2
CREDITS	**:**	2
HOURS PER WEEK	**:**	3 (Theory – Practice)
CONDITION	**:**	Mandatory

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, and arrays. Students develop computer programs applied for solving mathematical and physical problems.

III. COURSE OUTCOMES

At the end of the course, students:

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 / 4 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 / 10 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 blocks, “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 / 10 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. VECTOR AND MATRIX ARRAYS / 12 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.

V. LABORATORY EXPERIENCES

Lab 1: Data input and output
Lab 2: Decision instructions.
Lab 3: Repetitive instructions.
Lab 4: Vector arrays and selection sentences.
Lab 5: Algorithm programming

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. GRADING FORMULA

The final grade 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, 2012

2. **SCHILDT, Herbert**
   Turbo C Programming
   Osborne/McGraw-Hill, 2014