NATIONAL UNIVERSITY OF ENGINEERING
COLLEGE OF SCIENCES

COMPUTER SCIENCE PROGRAM

CC201 – OBJECT ORIENTED PROGRAMMING

I. GENERAL INFORMATION

CODE	**:**	CC201 Object Oriented programming
SEMESTER	**:**	3
CREDITS	**:**	4
HOURS PER WEEK	**:**	6 (Theory – Practice)
PREREQUISITES	**:**	CC102 Programming Languages
CONDITION	**:**	Compulsory

II. COURSE DESCRIPTION

At the end of this course, students will be able to solve medium-complexity problems through computer programs.

This course is made up of six subject units: classification and encapsulation; arrays, Inheritance, interfaces, polymorphism, multimedia, images, and application project.

III. COURSE OUTCOMES

1. Apply programming characteristics with classes and objects in the programs development.
2. Use the concept of heritance and polymorphism for the code reuse.
3. Implement methods for solving problems using arrays (vectors and matrices).
4. Carry out auto-documented and sorted programs development works.
5. Carry out research on Object-oriented Programming topics.

IV. LEARNING UNITS

1. CLASSIFICATION AND ENCAPSULATION / 18 HOURS
Classification / Classes and objects / Methods / Encapsulation / Access modifier / The reference “this” / Methods overloading / Instance member / Class members (Static).

2. ARRAYS, STRINGS / 36 HOURS
1-level array: vectors / Definition of vectors / Solution to vector problems: sorting, search and other problems / 2-level array: matrices / Definition of matrices /Solution to matrices problems / Objects array / Solution of problems involving object vectors and matrices / Strings / Solution to problems involving strings / Use of predefined library for the string work.

3. HERITANCE AND INTERFACES / 27 HOURS
Heritance: Concepts and application / Superclasses / Subclasses, hierarchy of classes / Heritance: Types / Simple heritance / Interfaces: Introduction to interfaces / Interfaces applications / Interface hierarchies.

4. POLYMORPHISM / 18 HOURS
Static polymorphism: Methods overloading, conductor overloading, methods superimposition / Dynamic polymorphism from methods superimposition / Interfaces and polymorphism.

5. MULTIMEDIA AND IMAGES / 18 HOURS

Multimedia: basic concepts / Image file formats / Multimedia: produced images creation and show / Images file show / use of images.

6. INTERMEDIATE-COMPLEXITY APPLICATION PROJECT / 9 HOURS

Execution and introduction of an intermediate-complexity application (team work).

V. LABORATORIES AND PRACTICAL EXPERIENCES:

Lab 1: Classification and encapsulation.
Lab 2: Arrays.
Lab 3: Heritance and interfaces.
Lab 4: Polymorphism.
Lab 5: Multimedia: Images.

VI. METHODOLOGY

The course is carried out in theory, lab and workshop sessions; there is one computer available for each student, and the instructor has a projector and a computer too. In theory sessions, the instructor introduces concepts and explains how lab guide problems should be solved. In lab and workshop sessions, Java Netbeans programming language is used for encoding the proposed solutions in theory sessions. Lab guides and offprints are published in the virtual campus. At the end of the course, students must hand in and expose a group application paper of intermediate complexity. In all sessions, feedback is required: answer to questions and constant reinforcements, encouraging students' active participation.

VII. EVALUATION FORMULA

The average grade PF is calculated as follows:

PF = 0.1*PA + 0.04*(G1 + G2 + G3 + G4 + G5) + 0.2*TA + 0.25*EP + 0.25*EF

EP: Mid-Term Exam	EF: Final Exam	TA: Application paper
PA: Active participation	G#: Course guides	

VIII. BIBLIOGRAPHY

1. **BOBADILLA, JESÚS**
   Java Through Examples (Spanish)
   Ra-Ma Editorial, 2006

2. **CEBALLOS, FRANCISCO JAVIER**
   Java 2: Programming Course
   Alfaomega Editorial, 2005

3. **DEITEL, HARVEY**
   Programming in Java
   Pearson Education, 2005