



**ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL**  
**Faculty of Electrical and Computer Engineering**  
**COURSE SYLLABUS**  
*Object Oriented Programming*

**1. CODE AND NUMBER OF CREDITS**

<b>CODE</b>	FIEC04622	
<b>NUMBER OF CREDITS: 4</b>	<b>Theoretical: 4</b>	<b>Practical: 0</b>

**2. COURSE DESCRIPTION**

This course covers the basic concepts and techniques for performing an analysis and design of solutions under the object-oriented paradigm, which is one of the most used paradigms in programming. Afterwards, the course proposes the use of the learned concepts to implement solutions to problems of medium complexity, using a language that applies this paradigm.

**3. PRE-REQUISITES AND CO-REQUISITES**

<b>PRE-REQUISITES</b>	FIEC04341 PROGRAMMING FUNDAMENTALS
<b>CO-REQUISITES</b>	

**4. CORE TEXT AND OTHER REQUIRED REFERENCES FOR THE TEACHING OF THE COURSE**

<b>CORE TEXT</b>	1. Deitel & Deitel. Java How to Program. 9th Edition, 2011, Prentice Hall.
<b>REFERENCES</b>	1. Y. Daniel Liang. Introduction to Java Programming, Comprehensive Version. 9th Edition, 2012, Prentice Hall. 2. Daniel Clark. Beginning C# Object-Oriented Programming. 2 edition, 2013, Apress. 3. Craig Larman. Applying UML and Patterns: An Introduction to Object-oriented Analysis and Design and Iterative Development. 3rd edition, 2004, Prentice Hall.

**5. COURSE LEARNING OUTCOMES**

At the end of the course, the student will be able to:

1. Apply the fundamental concepts of Object Oriented Programming (POO), such as objects, classes, abstraction, encapsulation, inheritance, and polymorphism, to solve problems.
2. Implement a GUI and use events to implement user requirements.
3. Implement solutions using the object-oriented paradigm, and an object-oriented programming language.
4. Design and implement UML diagrams.

**6. COURSE PROGRAM**

I.	Introduction to Object-Oriented Programming (1 session: 2 hours) <ul style="list-style-type: none"><li>• Introduction</li><li>• Frameworks</li><li>• Object-oriented programming advantages</li></ul>
II.	Objects and classes (3 sessions: 6 hours) <ul style="list-style-type: none"><li>• Data types</li><li>• Class definition: attributes and methods</li><li>• Access modifiers</li><li>• Objects and reference variables</li><li>• Variable scope</li><li>• Constructors</li><li>• Method overloading</li><li>• Libraries</li></ul>



	<ul style="list-style-type: none"><li>• Encapsulation: getters and setters methods</li><li>• Objects</li></ul>
III.	Object-oriented design – UML (4 sessions: 8 hours) <ul style="list-style-type: none"><li>• UML</li><li>• Use cases: specifications, diagrams, scenarios</li><li>• Class diagrams</li><li>• Interaction diagrams</li></ul>
IV.	Inheritance (2 sessions: 4 hours) <ul style="list-style-type: none"><li>• The inheritance relationship</li><li>• Constructors and inheritance</li><li>• Overriding</li><li>• Overriding vs. overloading</li></ul>
V.	Abstract classes and interfaces (1 session: 2 hours) <ul style="list-style-type: none"><li>• Abstract classes</li><li>• Interfaces</li></ul>
VI.	Polymorphism (2 sessions: 4 hours) <ul style="list-style-type: none"><li>• Dynamic binding</li><li>• Upcasting and downcasting</li><li>• Checking data types</li><li>• Examples</li></ul>
VII.	Static variables and methods (1 session: 2 hours) <ul style="list-style-type: none"><li>• Static variables</li><li>• Static methods</li><li>• Constants</li><li>• Wrapper Classes</li></ul>
VIII.	Exceptions and error handling (2 sessions: 4 hours) <ul style="list-style-type: none"><li>• Exception definition</li><li>• Exceptions types</li><li>• Exception handling</li></ul>
IX.	Object Persistence (1 sessions: 2 hours) <ul style="list-style-type: none"><li>• Introduction to XML</li><li>• Validation of XML Documents</li><li>• Object Serialization to XML</li><li>• Examples</li></ul>
X.	Unit testing (1 session: 2 hours)
XI.	Collections (2 sessions: 4 hours) <ul style="list-style-type: none"><li>• Definition</li><li>• Use of collections</li><li>• Examples</li></ul>
XII.	Graphic interface (3 sessions: 6 hours) <ul style="list-style-type: none"><li>• GUI Components (button, checkbox, list, etc.)</li><li>• Creating a user interface</li><li>• Event handling</li></ul>
XIII.	Graphical objects (2 sessions: 4 hours)
XIV.	Threads (1 session: 2 hours)
XV.	Laboratory (1 session: 2 hours)
XVI.	Review before the final exam

## 7. WORKLOAD: THEORY/PRACTICE

The term covers 28 sessions, 14 before midterm and 14 afterwards. Sessions are distributed as follows: 2



sessions per week of two hours each. Some sessions will be carried out in the laboratory.

### 8. CONTRIBUTION OF THE COURSE TO THE EDUCATION OF THE STUDENT

This course will provide students with the necessary knowledge of the object-oriented paradigm concepts (abstraction, encapsulation, polymorphism, etc.). It will also provide them with the ability to carry out an object-oriented analysis (use cases, scenarios, etc.), an object-oriented design (class diagrams, interaction diagrams, etc.) and the implementation of a solution under this paradigm using an object-oriented language. Student will also become familiar with the GUI and its components (windows, buttons, etc.).

BASIC TRAINING	PROFESSIONAL TRAINING	SOCIAL SKILLS DEVELOPMENT
	X	

### 9. THE RELATIONSHIP BETWEEN THE LEARNING OUTCOMES OF THE COURSE AND THE LEARNING OUTCOMES OF THE DEGREE PROGRAM

LEARNING OUTCOMES OF THE DEGREE PROGRAM*	CONTRIBUTIO N (High, Medium, Low)	LEARNING OUTCOMES OF THE COURSE**	THE STUDENT MUST:
a) Ability to apply computing and mathematical knowledge appropriate to their discipline.	---		
b) Ability to analyze a problem, identify and define the appropriate computational requirements for its solution.	MEDIUM	1, 4	Analyze a problem and, using the object-oriented paradigm, he/she should identify uses cases and scenarios. Design an appropriate class diagram for the defined solution.
c) Ability to design, implement, and evaluate a computer-based system, processes, components and programs that meet specific needs	HIGH	1, 2, 3, 4	Create classes and implement either console or graphical applications that instantiate these classes.  Collaboratively implement an application, based on a previous design.  Collaboratively implement a graphical application with an object-oriented language.
d) Ability to function effectively in teams to achieve a common goal.	MEDIUM	2, 3	Collaboratively implement an application, based on a previous design.  Collaboratively implement a graphical application with an object-oriented language.
e) Comprehension of professional, ethic, legal security and social responsibilities of their profession	---		
f) Ability to communicate effectively with a range of audiences.	---		
g) Ability to analyze the local and global impact of computing on individuals, organizations and society	---		
h) Acknowledge the need for and ability to engage in continuous professional development.	---		
i) Ability to use techniques,	HIGH	2, 3	Appropriately interpret the API



skills, and current tools necessary for the practice of computing.			documentation of an object-oriented language.  Use software tools to develop, debug and generate the documentation of a software application.
j) Capacity to lead, manage or undertake projects.	---		

**10. EVALUATION IN THE COURSE**

Evaluation activities	
Exams	x
Tests	x
Homework/tasks	x
Projects	X
Laboratory/Experiments	
Class participation	
Visits	
Other	

**11. PERSON RESPONSIBLE FOR THE CREATION OF THE SYLLABUS AND THE DATE OF ITS CREATION**

Created by	Ing. Jorge Rodriguez E.
Date	11 MAR 2013

**12. APPROVAL**

ACADEMIC SECRETARY OF THE ACADEMIC DEPARTMENT	DIRECTOR OF TECHNICAL ACADEMIC SECRETARY
NAME: Sra. Leonor Caicedo G.	NAME: Ing. Marcos Mendoza V.
SIGNATURE:	SIGNATURE:
Date of approval by the Directive Council: 2013-334 2013-08-12	<i>Ing. Marcos Mendoza V.</i> DIRECTOR DE LA SECRETARIA TÉCNICA ACADÉMICA

**13. VALIDITY OF THE SYLLABUS**

RESOLUTION OF THE POLYTECHNIC BOARD:	13-10-269
DATE:	2013-10-17