

University of Puerto Rico
 Mayagüez Campus
 College of Engineering
 Department of Electrical and Computer Engineering
 Bachelor of Science in Computer Engineering

Course Syllabus

1. General Information:	
Alpha-numeric codification: ICOM 4015 Course Title: Advanced Programming Number of credits: 4 Contact Period: 3 hours of lecture and 2 hours of laboratory per week Required in ICOM	
2. Course Description:	
English: Advanced programming techniques applied to the solution of engineering problems;extensive use of subprograms,logical and specifications statements. Principles of multiprogramming, multiprocessing,and real-time systems.	
Spanish: Técnicas Avanzadas de Programación Aplicadas A la Solución de Problemas de Ingeniería: Uso Amplio de Subprogramas, de Expresiones Lógicas y de Especificación. Principios de Multiprogramación, Multiprocesamiento y Sistemas de Tiempo Real.	
3. Pre/Co-requisites and other requirements:	
Pre-requisite: INGE 3016	
4. Course Objectives:	
The student will learn how to use sub-routines, arrays, classes, and other object-oriented techniques for the design, implementation and analysis of complex software systems used in Computer Science, Computer Engineering and Software Engineering.	
5. Instructional Strategies:	
<input checked="" type="checkbox"/> conference <input type="checkbox"/> discussion <input checked="" type="checkbox"/> computation <input checked="" type="checkbox"/> laboratory <input type="checkbox"/> seminar with formal presentation <input type="checkbox"/> seminar without formal presentation <input type="checkbox"/> workshop <input type="checkbox"/> art workshop <input type="checkbox"/> practice <input type="checkbox"/> trip <input type="checkbox"/> thesis <input type="checkbox"/> special problems <input type="checkbox"/> tutoring <input type="checkbox"/> research <input type="checkbox"/> other, please specify:	
6. Minimum or Required Resources Available:	
Students will use the Departmental computer laboratories to complete course projects.	
7. Course time frame and thematic outline	
Outline	Contact Hours
Introduction: Discussion of the organization of the course, general view of the topics to be covered, and ethical behavior of the computing professional.	1
Introduction to Computing: Introduction to the computer language to be used, and how to get it.. General format of a computer program, compilation and execution of a program, and recognition of errors.	1.5
Fundamental Data Types, Operations, Expressions, and I/O: The numeric data types, strings, characters, and booleans. Variables, constants and literals of each of these primitive data types. The assignment statement, mathematical expressions, mathematical functions, type conversions, and comparison between primitive data types and objects. The String data type. Text based I/O.	2.5
Decisions: Review the topic of decisions in programs, this time in the context of the programming language used. We study selection constructs, the <i>if-statement</i> and the <i>switch-statement</i> . We also study about Boolean expressions and grouping of statements in a program.	1.5
Iteration: Review the topic of iterations in programs, the <i>while</i> and <i>for loops</i> .	1.5
Functions: Review of functions in a program, by-reference and by-value parameters, returning values, the activation record. Use of functions in procedural	3

abstractions.	
Arrays: Introduction to arrays – one-dimension and two-dimensional arrays. Different applications and typical operations with these structures are discussed.	3
Introduction to ADTs: Study of the concept of ADT and their application in describing new data types. The following ADTs are studied: Strings and lists. Different applications are discussed.	3
Object-Oriented Concepts: Study of objects, classes, interfaces, inheritance, virtual functions, polymorphism, and language constructs for their support. Emphasis is given to their application in the implementation of ADT's.	4
Object Oriented Design: Study of the object-oriented techniques applied to the software development process. In particular: the software life cycle, discovery of classes and member functions, cohesion and coupling, UML class diagrams, modularization, and the development of complex programs.	4
Input/Output and File Management: File operations, streams, conversions, sequential files, and random access files. Application in databases.	3
Recursive Algorithms: Study of recursion, the implementation mechanisms commonly used in programming languages, and its use in deriving algorithmic solutions to problems. Particular cases are studied: iteration by recursion and exhaustive searches.	3
Sorting Algorithms: Sorting algorithms for array-based containers; selection-sort, merge-sort, and quick-sort. Comparison of performance.	4
Searching Algorithms: Study of search algorithms over data containers based on arrays – sequential and binary search.	3
Introduction to the Analysis of Algorithms: Complexity issues in data structures and algorithms: correctness, execution time, and space requirements. Asymptotic notation. Particular algorithms are analyzed in sorting and searching.	4
Exams and discussions	3
Total hours: (equivalent to contact period)	45

8. Grading System

Quantifiable (letters) Not Quantifiable

9. **Evaluation Strategies** (Suggested): The faculty member teaching the course will provide the student with the evaluation strategy he/she will be using throughout the semester. This will be done within the first week of classes.

	Quantity	Percent
<input checked="" type="checkbox"/> Exams	3	35%
<input checked="" type="checkbox"/> Final Exam	1	30%
<input checked="" type="checkbox"/> Short Quizzes		
<input type="checkbox"/> Oral Reports		
<input type="checkbox"/> Monographies		
<input type="checkbox"/> Portfolio		
<input checked="" type="checkbox"/> Projects	4	25%
<input type="checkbox"/> Journals		
<input checked="" type="checkbox"/> Other, specify: Laboratory	10	10%
TOTAL:		100%

10. Bibliography:

Cay Horstmann, *Java Concepts* (5th Edition) John Wiley.

Paul T.Tymann and G. Michael Schneider, *Modern Software Development Using Java*, Thomson Brooks/Cole, 2004

11. According to Law 51

Students will identify themselves with the Institution and the instructor of the course for purposes of assessment (exams) accommodations. For more information please call the Student with Disabilities Office which is part of the Dean of Students office (Chemistry Building, room 019) at (787)265-3862 or (787)832-4040 extensions 3250 or 3258.

12. Contribution of Course to meeting the requirements of Criterion 5:

Math	Basic Science	General	Engineering Topic
			√

13. Course Outcomes**Map to Program Outcomes**

1. Knowledge of the contemporary issues in discipline of Computer Science and Engineering (a)
2. Ability to apply knowledge of math in the analysis and design of fundamental algorithms. (a)
3. Knowledge of the programming process. (a)
4. Ability to apply abstraction techniques including structured naming, procedures, encapsulation, classes and polymorphism to the design and development of complex software systems. (e)
5. Use of a modern structured programming language. (a)
6. Ability to apply good programming style practices. (a)
7. Ability to use a software development toolkit: editor, compiler and debugger, to implement software modules or a complete system within the allotted time specified by deadline for a deliverable. (k)

Person (s) who prepared this description and date of preparation: Pedro Rivera. Submitted by: Manuel Rodríguez, March 2007