Introduction to Computer Programming in Engineering and Science

Objectives:	00UV	**Discipline:**	Computer Science, Physics, Mathematics, Biology, Chemistry
Ponderation:	3-2-3	**Course Code:**	360-420-DW
Prerequisite:	00UM, 00UP, 00UT	**Course Credit:**	2 2/3
Corequisite:	00UQ	**Semester:**	4

Introduction

Introduction to Computer Programming in Engineering and Science is offered to students in their final semester. The course focuses on the use of computational methods to solve complex problems in science and

Objectives and Standards for

OBJECTIVE	STANDARD	LEARNING OBJECTIVES
	4. Correct graphical and mathematical representations of systems 5. Appropriate use of	

OBJECTIVE	STANDARD	LEARNING OBJECTIVES

OBJECTIVE	STANDARD	LEARNING OBJECTIVES
		4.9.8. Identify the order of precedence of relational and logical operators.
		4.9.9. Understand short circuit evaluation.
		4.9.10. Understand data validation and its importance to data integrity and system.
	4.10. Design a method that solves a single identifiable task	4.10.1. Solve problems by constructing and using methods that perform one major task.
		4.10.2. Create a method.
		4.10.3. Invoke a method.
		4.10.4. Pass primitive data values to a method.
		4.10.5. Return a single value from a method.
		4.10.6. Pass a reference value to a method.
5. To apply numerical methods and computer programming techniques to solve scientific and engineering problems	5.1. To solve mathematical problems numerically using Java	5.1.1 Use arithmetic operators respecting the rules of precedence.
		5.1.1. Convert arithmetic expressions into Java expressions.
		5.1.2. Evaluate Java arithmetic expressions.
		5.1.3. Recognize the limitations of numeric data types.
		5.1.4. Explain overflow and underflow errors.
		5.1.5. Cast from one primitive data type to another.
		5.1.6. Explore and use classes from Java's standard packages.
		5.1.7. Use methods of the Math class – pow(), sqrt(), round(), random(), min(), and max().
	5.2. Learn appropriate numerical methods to computation of a model	5.2.1. Define appropriate convergence conditions for iterative methods.
		5.2.2. Understand and minimize errors related to numerical solution of mathematical equations.
		5.2.3. Solve scientific models involving non-linear equations by a variety of methods.
		5.2.4. Solve scientific models involving first-order ordinary differential equations (i.e. initial and boundary value problems) using Euler or related methods.
		5.2.5. Solve other type of scientific models using the appropriate numerical techniques (e.g. by Fourier or Laplace transform, Monte-Carlo, Newton-Raphson, etc.)
		5.2.6. Perform simple numerical optimization given a well-posed problem (e.g. by golden section search method, downhill simplex method, genetic algorithm, etc.)
6. To apply a general approach to model development, application and		

6.2. Appropriate use of computational methods	6.2.1. Translate model into a form suitable for a computational solution
	6.2.2. Identify numerical method(s) best suited to solve a given model
	6.2.3. Identify which optimization algorithm is best suited to solve a model
	6.2.4. Determine appropriate numerical parameters for solution of the model (*e.g.* sample size, time step, *etc.*)
	6.2.5. Evaluate a solution for convergence
6.3. Evaluation of computational methods	6.3.1. Test the validity of the computational method on simpler problems that are well understood.
	6.3.2.