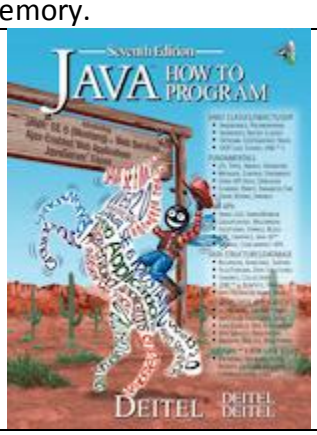
Applied Information Technology Department

Course Syllabus

IT208 Program Design and Data Structures

Revised 1/16/2009

Instructor Information:	***Irene E. Bruno, PhD*** Phone: 703-993-8541 Email: ibruno@gmu.edu Office: Prince William Campus, 102E Bull Run Hall
Course Catalog Description:	Fundamentals of data structures and analysis of algorithms. Large programs written in a modern, high-level programming language. Stresses abstraction, modular design, code reuse, and correctness.
Prerequisite:	The prerequisite for this course is IT108 (or an approved equivalent course). A grade of "C" or better **must** be achieved in the prerequisite course **before** a student is qualified to take this course. The prerequisite course must be completed prior to, no concurrently with, this course. This requirement will be **strictly enforced**.
Rationale:	Programming is an essential skill for IT students and IT professionals. Understanding how a computer is instructed to accomplish tasks leads to an appreciation of the underlying concepts of the Information Technology discipline. Learning how to solve a problem using a procedural or object-oriented programming language provides a strong foundation that will be used in Database, Security, Web Development and Networking courses.
Overall Educational Objective	To further students' problem solving skills using a high level, object-oriented programming language as a tool. To provide students with the skill to analyze the efficiency of algorithms based on time and memory.
Textbook:	Deitel (2007). Java: How to Program (7th ed). ISBN-10: 0132222205 ISBN-13: 9780132222204. Publisher: Prentice Hall Copyright: 2007
	
Course Coordinator:	Irene E. Bruno, PhD Assistant Professor

Course Learning Outcomes:	• Analyze and explain the behavior of complex programs involving the fundamental programming concepts: both procedural and object-oriented • Apply the techniques of structured (functional) decomposition to break a sizeable program into smaller pieces • Describe the concept of recursion and give examples of its use. • Implement and trace the execution stack of a simple recursive function. • Discuss and identify the concepts of encapsulation, abstraction, inheritance, and polymorphism. • Describe how the class mechanism supports encapsulation and information hiding. • Design, implement, and test the implementation of "is-a" relationships among objects using a class hierarchy and inheritance. • Compare and contrast the notions of overloading and overriding methods in an object-oriented language. • Utilize iterators to access the elements of a container. • Describe how constructors and destructors relate to the life of an object. • Describe the relationship between an object and its corresponding class. • Identify the necessary properties of good algorithms. • Develop code that responds to exception conditions raised during execution. • Explain the difference between event-driven programming and command-line programming. • Design, code, test, and debug simple event-driven programs that respond to user events.
Assessment of Outcomes:	Final Exam 40% Midterm Exam 25% Programming Assignments 25% Project 10%
Grading:	The grading scale for this course is (Raw Scores may be adjusted at the instructor's discretion) : 98 – 100 % A + 92 – 97 % A 90 – 91 % A- 88 – 80 % B+ 82 – 87 % B 80 – 81 % B- 78 – 79 % C+ 72 – 77 % C 70 – 71 % C- 60 – 69 % D 0 – 59 % F

Honor Code:	The George Mason University Honor Code will be strictly enforced in this course. For this course, the following requirements are specified: •All assessable work is to be prepared by the individual student, unless the Instructor explicitly directs otherwise. •All work must be newly created by the individual student for this course for this semester. Any usage of work developed for another course, or for this course in a prior semester, is strictly prohibited without prior approval from the instructor. Students may seek assistance with assigned work, (and are encouraged to do so) provided: •The directions for the assigned work do not prohibit such assistance. •Such assistance is acknowledged in the submitted work, clearly identifying the person(s) giving assistance and the nature of the assistance given. •Any work to be submitted is prepared entirely and exclusively by the student submitting it. Students are expressly prohibited from sharing any assessable work for this course in any manner with other students
Additional Information:	Late assignments are NOT accepted in this course. Students may submit incomplete assignments for potential partial credit but extensions will not be approved. Exams are closed notes and closed book. A computer will not be accessible. Students must arrive promptly for exams. Students arriving after the exam start time may not be permitted to sit for the exam. Further information on the course schedule, specific requirements and assignments may be found on the Blackboard site to which all students are given access. Students are expected to communicate with their instructor and/or teaching assistant in person, via office phone or Mason email. Students are expected to attend each class and complete any preparatory work (including assigned reading) as well as participate actively in lectures and discussions.
Course Schedule:	Will be posted on the Blackboard site