

Date: October 2005

Course Name: Introduction to Java
Prefix and Number: CSC 228
Credit Hours and Contact Hours: 3 credit hours – 4 contact hours

Course Description:

This course is designed to present to the student the basic data structures necessary to design and write structured programs. The topics covered include procedures and modules, functions, arrays, records, strings, and files. Prerequisite: Successful completion of CSC115 with a C or better.

II. Course Outcomes and Objectives

Learning Outcomes

Upon completion of the course the participant will be able to:

- A. Develop an algorithmic approach to problem solving. Flowchart and document any algorithm.
- B. Write programs in a high level computer language to solve these problems defined in part E., above. Enter the programs into the computer, compile and thoroughly test them. Prepare written internal and external documentation that explains the purpose and function of each program and its input and output.
 - Programming topics
 - syntax
 - identifiers and data types
 - assignment statements and hierarchy of operations
 - input/output statements and format
 - block
 - selection: boolean, if-then-else
 - looping: count controlled, event controlled
- C. design and construct programs with procedures/modules and to manipulate variables using structured programming techniques.
- D. design and construct programs with appropriate functions including recursion.
- E. use enumerated and user defined data types in structured programs.
- F. construct and use one and two-dimensional arrays. They will be able to understand and use searching techniques, parallel arrays, merging techniques, and sorting techniques.
- G. construct, and use appropriately, records in structured programs.
- H. use files as an appropriate data structure.

Relationship to Academic Programs and Curriculum

This course is not currently required for any of the three majors, but will be highly recommended for students looking into programming or web development. It may eventually replace C++ as our programming language.

Competencies Addressed in this Course

Writing
Oral Communication
Reading
Computer Literacy
Problem Solving and Critical Thinking
Ethics and Values
Global Concerns
Professional Competency

III. Methods of Instruction

Types of Course Materials

1. Textbook: A JAVA programming text

Methods of Instruction

1. Lecture
2. Discussions
3. Demonstrations
4. Group programming
5. Programming

Assessment Measures

Activities will emphasize problem solving using the computer - specifically programming the computer. The students should demonstrate proficiency programming in the JAVA language.

Methods of Evaluation

The demonstration of the satisfactory achievement of the above learning outcomes will be the responsibility of the student, facilitated by the instructor. The department maintains a very open attitude and believes each instructor should determine the grading system and evaluation methods that will be used in his/her sections of the course. It is highly recommended that these be communicated to the students the first week of the semester, preferably in writing.

Among the evaluation methods that could be used are exams, quizzes, and programming assignment projects and programs. Any grading system used must be consistent with the College Catalog and Middle States grading procedure. Course policies about attendance, late work, plagiarism, etc.. are at the discretion of the instructor. If such policies exist, they must be communicated to the student, preferably in writing.

IV. General Outline of Topics Covered

A. Overview of Programming

- a. What is a Program
- b. Ethics
- c. Problem Solving Techniques
- B. JAVA syntax and semantics**
 - a. Data Types
 - b. Program Entry
- C. Event-Driven Output**
 - a. Classes and Methods
 - b. Formatting output
 - c. Testing and debugging
- D. Numeric types and expressions**
 - a. Overview of JAVA data types
 - b. Numeric data types
 - c. Arithmetic Expressions
 - d. Strings
- E. Software Design**
 - a. Getting data into programs
 - b. Creating buttons, fields, listener
 - c. Object-oriented design
- F. Conditions, Logic Expressions and Selection**
 - a. Flow of Control
 - b. If statements
- G. Classes and Methods**
 - a. Encapsulation
 - b. Class Syntax
 - c. Packages
- H. Inheritance, Polymorphism, and Scope**
- I. File I/O and Looping**
- J. Additional Control Structures**
 - a. Switch
 - b. Do
 - c. For
 - d. Try-catch-finally statement
- K. One-Dimensional Arrays**