

Course Number, Course Title and Semester Hours

CSC 130, Computer Science I (Fall 2009), 4 sh.

Course Coordinator

Shannon Duvall

Current Catalog Description

This introduction to programming and problem solving emphasizes applications from quantitative disciplines and incorporates in class programming experiences. Offered fall and spring.

Textbook

Absolute Java, 4th Edition, by Savitch

References

Java Sun API : Information on all standard Java classes. Students are encouraged to look up information on any needed code and to use any classes in the standard toolkit, even if not discussed in class. <http://java.sun.com/javase/6/docs/api/>

Eclipse Homepage: Students can download and have access to support for the preferred course IDE. <http://www.eclipse.org/>

Course Outcomes

1. Understand the basics of problem-solving and creating algorithms to solve simple problems
2. Explain the behavior of simple fundamental Java programs
3. Implement, test, and debug simple programs involving I/O, conditionals, iterative structures, and methods
4. Implement, test, and debug simple programs involving a collection
5. Know how to use an API
6. Know the basics of Java and object-oriented programming
7. Learn the ACM Code of ethics and be able to make decisions based on this and other prevalent ethical codes
8. Learn to work on programming projects in groups

Relationship between Course Outcomes and Program Outcomes

1. Understand the basics of problem-solving and creating algorithms to solve simple problems (Outcomes 1, 9)
2. Explain the behavior of simple fundamental Java programs (Outcomes 1, 9)
3. Implement, test, and debug simple programs involving I/O, conditionals, iterative structures, and methods (Outcomes 1, 3, 9)
4. Implement, test, and debug simple programs involving a collection (Outcomes 1, 3)
5. Know how to use an API (Outcome 9)
6. Know the basics of Java and object-oriented programming (Outcomes 3, 9)

7. Learn the ACM Code of ethics and be able to make decisions based on this and other prevalent ethical codes (Outcome 7)
8. Learn to work on programming projects in groups (Outcome 4)

Prerequisites by Topic

- MTH 100: Algebra skills

Major Topics Covered in the Course

1. Programming Fundamentals/Fundamental Constructs (7)
 - a. Analyze and explain the behavior of simple programs involving the fundamental programming constructs of Java
 - b. Modify and expand short programs that use standard conditional and iterative control structures and methods
 - c. Design, implement, test, and debug a program that uses each of the following fundamental programming constructs: basic computation, simple I/O, standard conditional and iterative structures, and the definition of methods
 - d. Choose appropriate conditional and iteration constructs for a given programming task
 - e. Apply the techniques of structured (functional) decomposition to break a program into smaller pieces
 - f. Describe the mechanics of parameter passing
2. Programming Fundamentals/Algorithmic Problem Solving (5)
 - a. Discuss the importance of algorithms in the problem-solving process
 - b. Identify the necessary properties of a good algorithm
 - c. Create algorithms for solving simple problems
 - d. Use Java to implement, test, and debug algorithms for solving simple problems
 - e. Describe strategies that are useful in debugging
3. Programming Fundamentals/Data Structures (3)
 - a. Discuss the use of primitive data types and built-in data structures
 - b. Write programs that use arrays and strings
4. Programming Languages/Declarations and Type (1)
 - a. Discuss type incompatibility
5. Programming Languages/Object-Oriented Programming (2)
 - a. Design, implement, test, and debug simple programs in an object-oriented language
6. Social and Professional Issues/Social Context (2)
 - a. Describe positive and negative ways in which computing alters the modes of interaction between people
 - b. Articulate the impact of the input deficit from diverse populations in the computing profession
7. Social and Professional Issues/Professional Ethics
 - a. Evaluate the professional codes of ethics from the ACM, IEEE Computer Society, and other professional organizations
8. Software Engineering/Using APIs (1)

- a. Explain the value of application programming interfaces (APIs) in software development
 - b. Use class browsers and related tools during the development of applications using APIs
9. Software Engineering/Tools and Environments (1)
- a. Demonstrate the capability to use a range of software tools in support of the development of a software product of medium size

Estimate Curriculum Category Content in semester hours. Maximum number of semester hours per course is 4. Times should be in increments of .5. (Advanced is a topic requiring previous core materials – most likely found in a 300 or 400 level class.)

Category	Core	Advanced
Data Structures	.5	
Algorithms	2	
Software Design	.5	
Computer Architecture		
Programming Languages	1	

Course Assessment (List date of pre or post assessment)

Proposed changes from last offering (Pre assessment)

In keeping with last year’s proposed changes, this year I will be doing more strict assessment of in-class work to make sure that computer time in class is spent on task. I will also be giving more individual, out-of-class assignments to assure that every person is required to learn the material.

Learning outcomes to be assessed, how they will be assessed and success criteria (Pre assessment)

This semester I will be assessing two learning objectives:

- 1. Understand the basics of problem-solving and creating algorithms to solve simple problems
- 2. Explain the behavior of simple fundamental Java programs

I feel these two objectives are the most indicative of Program Outcome 1: An ability to apply knowledge of computing and mathematics appropriate to the discipline. Being able to explain the behavior of given code and to write new code that solves a problem are the two most essential abilities to apply knowledge of computer science.

Since I am most interested in the individual’s ability to analyze and write code, I will base the assessment on individual programs and quizzes/tests only. Success is defined as an average of at least 75% on individual programs written and questions on the two exams that involve explaining the behavior of given code.

Assessment data and analysis (Post assessment)

Proposed changes for next offering (Post assessment)