COMPUTER SCIENCE 161

INTRODUCTION TO COMPUTER SCIENCE

I. Introduction

A. Catalog Description

This course is an introduction to computer science and programming. The programming language Java is used to illustrate concepts in computer science. The course emphasizes the use of the computer as a problem solving tool and the development of good programming style. CSCI 161 is the introductory course for students planning to major or minor in computer science. Students planning on taking further courses in computer science should select this course instead of CSCI 158. *Prerequisite: Three years of high school mathematics, MATH 110 or its equivalent.* Satisfies the Mathematical Approaches core requirement.

B. Objectives

1. First Course for Computer Science Majors - CSci 161 meets the objectives for a first course in computer science as recommended by the ACM. These objectives include the introduction of problem solving methods and algorithm development; the teaching of a high-level language that is widely used, and to teach how to design, code, debug and document programs using techniques of good programming style.

2. Service Course - CSci 161 satisfies the Mathematical Approaches requirement within the University's core curriculum. CSci 161 is designed to teach computer science and programming while emphasizing analysis and problem solving, enabling the student to write software in support of a variety of majors including those within the sciences, mathematics and business.

This course satisfies the Mathematical Approaches category of the university's core curriculum by developing an appreciation of the power of Computer Science and formal methods to provide a way of understanding a problem unambiguously, describing its relation to other problems, and specifying clearly an approach to its solution. A student in this course will develop a variety of mathematical skills, an understanding of formal reasoning, and a facility with applications. Specifically, this course will provide the student with the ability to analyze a problem, to design a systematic way of addressing that problem (an algorithm), and to implement that algorithm in a computer programming language.

C. Prerequisites

To take CSci 161, a student would be expected to have certain minimal quantitative skills. Three years of high school mathematics, Math 110 or its equivalent is required for admission into CSci 161.

II. Required Topics

The following outline represents the "core" material for a CSci 161 course. Each course should cover, at a minimum, these topics.

A. Computer Organization

1. An overview identifying components and their functions, including main and auxiliary memory.
2. A brief introduction to the language of a machine.
3. An understanding of Java's role in developing applications for the Internet and World Wide Web.

B. Problem-solving and Algorithm Development

1. Problem analysis.
2. Object-oriented design with basic UML.
3. Program design including the use of basic control structures and structured programming principles.
4. Testing and debugging techniques.

C. Programming Techniques

1. Primitive data types; constants and variables including integers, doubles, booleans, and characters.
2. Programming calculations; assignment statements, expressions and arithmetic operators, casting and standard functions.
3. Objects, including encapsulation and visibility modifiers; String and object references, class and method definitions.
4. Inheritance, polymorphism, abstract classes and interfaces.
5. Binary arithmetic; internal data representation for integers and characters.
6. Input-output including stream I/O, reading and writing data to/from a text file.
7. Control structures; sequence, selection, iteration.
8. Procedure and function methods, parameter/argument transmission, overloading, and local variables.
9. Elementary data structures including arrays, arrays as parameters and arrays of objects.
10. Two dimensional arrays
11. Elementary sorting and linear searching.

III. Optional Topics

1. Exceptions and Exception Handling
2. Recursion
3. An introduction to HTML and applets
4. Reading/writing binary files
6. An introduction to Swing GUI programming; including frames, layout managers, and event-driven programming.
7. Binary searching

III. Bibliography

Y. Daniel Liang, Introduction to Java Programming, Prentice-Hall.

H. M. Deitel and P. J. Deitel, Java: How to Program, Prentice-Hall.

J. Lewis and W. Loftus, Java Software Solutions: Foundations of Program Design, Addison-Wesley.

D. Arnow and G. Weiss, Introduction to Programming Using Java: An Object-Oriented Approach, Addison-Wesley.

C. Horstmann, Computing Concepts with Java 2 Essentials, John Wiley & Sons.

R. Morelli, Java, Java, Java: Object-Oriented Problem Solving, Prentice-Hall.

J. Slack, Programming and Problem Solving with Java, Brooks/Cole.

C. Thomas Wu, An Introduction to Object-Oriented Programming with Java, WCB/McGraw-Hill.

J. Nino and F. Hosch, Introduction to Programming and Object-Oriented Design Using Java, John Wiley & Sons.