

CSE 21 INTRODUCTION TO COMPUTER SCIENCE I
(Required for CSE)

| | |
|--------------------------------|---|
| Catalog Data: | CSE 21: Introduction to Computer Science I (Credit Units: 6) Introduces fundamental concepts related to computer software design and construction. Develops initial design and programming skills using a high-level language. Fundamental concepts of control structures, data structures, and object-oriented programming. Same as ICS 21. Only one course from CSE21/ICS 21 and ICS H21 may be taken for credit. May not be taken for credit after Informatics 42. (V) (Design units: 0) |
| Textbook: | Horstmann, Cay. <i>Java Concepts</i> , 4 th Edition, Wiley & Sons, 2005. |
| References: | Additional reference materials are on reserve in the Science Library. |
| Coordinator: | Amelia Regan |
| Course Outcomes: | Students will: Write programs in Java using: Basic statements (assignment, I/O) Basic data types (int, long, double, Boolean) Selection (if, switch) Loops and loop control (for, while, do; the dangers of break) Methods/function (defining, calling, parameter passing) Demonstrate proficiency in exception handling as applied to files and scoping rules. Identify the appropriate uses of data structures and be able to read code to determine whether they are properly and reasonably implemented. Write code to implement conventional algorithms in that context. Understand the fundamental concepts of Object Oriented Programming |
| Prerequisites by Topic: | |
| Lecture Topics: | Introduction to the course; compiling, linking and executing Java programs (1.5 hours) A simple Java program: data abstraction; classes and fields; objects; defining and calling methods; public vs. private; assignment and print statements; importing; parameter passing; return values; local variables (4.5 hours) Java basics in detail (1.5 hours) Decision statements (1.5 hours) Loops and simple loop algorithms (1.5 hours) Interclass communication; value and reference parameters; introduction to testing (1.5 hours) Basic ArrayLists (1.5 hours) Exception handling (1.5 hours) File handling (3.0 hours) Parallel and nested ArrayLists (3.0 hours) Recursion (1.5 hours) Inheritance and polymorphism (1.5 hours) Interfaces; introduction to applets and graphics (1.5 hours) Introduction to event handling and graphical user interfaces (GUIs) (1.5 hours) |

Class Schedule: Meets for 3 hours of lecture, 6 hours of lab, and 2 hours of discussion each week for 10 weeks.

Computer Usage: Windows XP, Textpad, Java

Laboratory Projects:

Lab #1. Write a Java program that 1) accepts some test scores from the user, 2) computes some statistics from these scores and 3) displays the entered scores and the computed statistics on the console window.

Lab #2. Enhance the test program to make it more useful, and to check input data for (some kinds of) errors; along the way, you apply a number of important Java concepts and get practice with several Java statements.

Lab #3. A serious collector of early recorded music has left his collection to a music archive. You've been hired to create a title index and some "media counts" for the collection from a file that lists the donated items. The archive needs the title index to easily find items; it needs the counts to determine how many storage shelves it needs for each kind of media.

Lab #4. Users of your music archive program (the one you wrote for Lab #3) have noticed that it takes a fair amount of time to produce the index file. You've been directed to speed up the program. Use a List ADT and implement it using linked lists with iterators.

Lab #5. Recall that you were directed to modify your music archive program so that it produces the music index file more quickly. This requirement was to be met by using a linked list (instead of an ArrayList) to store music items. The list class was written in Lab #4; revise your music program (Lab #3) to use it.

Professional Component: Contributes towards the Computer Science and Engineering major requirements for Engineering and Computer Topic courses.

Relationship to Program Outcomes: This course relates to Program Outcomes [REDACTED] as stated at:
<http://undergraduate.eng.uci.edu/degreeprograms/compsci/mision>

Design Content Description

Approach:

Lectures:

Laboratory Portion:

Grading Criteria:

| | |
|--------------|------|
| Midterm exam | 40% |
| Final exam | 60% |
| | 100% |

Lab Assignments (Pass/Fail but each lab must be passed in order to pass the class)

Estimated ABET Category Content:

Mathematics and Basic Science: 0 credit units or 0%
Computer Science: 6 credit units or 100%
Engineering Science: 6 credit units or 100%
Engineering Design: 0 credit units or 0%

Prepared by: Amelia Regan **Date:** July 2007

CEP Approved: Fall 2003