Las Positas College
3000 Campus Hill Drive
Livermore, CA 94551-7650
(925) 424-1000
(925) 443-0742 (Fax)

Course Outline for CS 31

JAVA PROGRAMMING

Effective: Fall 2003

I. CATALOG DESCRIPTION:
CS 31 — JAVA PROGRAMMING — 4.00 units

Applications programming in Java for students already familiar with the concepts of programming. Topics will include in Applets and Swing, multimedia, presenting data files over the web, elementary data structures (queues, linked list, stacks) and vectors, binary searching, sorting, JDBC (Java Data Base Connectivity), Remote Method Invocation (RMI), and Java Beans.

3.00 Units Lecture 1.00 Units Lab

Prerequisite
CS 1 - Computing Fundamentals I
or

-
with a minimum grade of C

Grading Methods:
Letter or P/NP

Discipline:

	MIN
Lecture Hours:	54.00
Lab Hours:	54.00
Total Hours:	108.00

II. NUMBER OF TIMES COURSE MAY BE TAKEN FOR CREDIT: 1

III. PREREQUISITE AND/OR ADVISORY SKILLS:
Before entering the course a student should be able to:

A. CS1

IV. MEASURABLE OBJECTIVES:
Upon completion of this course, the student should be able to:

A. GENERIC: These outcomes are being developed throughout the entire programming sequence. Upon completion of the course, to an intermediate level, students should be able to: Programming Skills
   1. present the elements and features of the development environment;
   2. explain and use the design process;
   3. define and use functions and storage classes;
   4. define and explain trends in programming standards;
   5. write, compile, test and debug programs;
   6. present the characteristics of object-oriented programming;
   7. define and use data types and variables;
   8. define and use multi-dimensional arrays;
   9. define and use constructor functions;
   10. define and use function overloading;
   11. define and use inheritance mechanisms in OOP;
   12. define and use user interfaces;
   13. define and use file I/O;
   14. define and develop class modules;
   15. develop and use event-driven programs;

B. Systems Analysis
   1. develop high-level systems and functional specifications;
   2. define general scope of work to meet requirements and constraints;

C. Systems Design
   1. specify major subsystems and interfaces;
   2. develop detail design specifications;
   3. select design methodology and tools;

        4. identify maintenance requirements;
    D. Technical Documentation
        1. write in a concise and precise form appropriate for technical documentation;
        2. explain and use the processes and techniques of technical documentation;
        3. record system specifications accurately and completely;
    E. Testing and Debugging
        1. select debugging and testing methodology, and develop comprehensive and systematic test plan;
        2. develop testing procedures;
        3. conduct tests in the most efficient way;
        4. test programs, and document errors and solutions;
    F. User Interface Design
        1. define the requirements for the user interface;
        2. detail the development process and methods best suited for the project;
        3. develop user interface (UI) to meet user requirements;
        4. test Uis;
    G. Problem Solving
        1. recognize a wide range of problems, and assess their impact on the system;
        2. use a wide range of troubleshooting methods and tools to isolate problems;
        3. select the appropriate approach to identify causes of the problem based on the given situation.
    H. SPECIFIC: These outcomes are detailed specifically for this course. Upon completion of the course students should be able to: write programs using the basic grammar and syntax of Java;
    I. write programs that create queues, linked list, stack, tree and hash tables;
    J. write programs using different sorts: bubble, insert, shell, selections, and quick;
    K. write programs using the binary search algorithm;
    L. write programs projecting MIDI and WAV sound files over the web and within Swing;
    M. write programs accessing sequential and random files within Swing and from an Applet;
    N. write programs using Access databases and ODBC;
    O. write programs using Swing with Access databases;
    P. write programs using Java Beans;
    Q. write programs involving a network;
    R. define and work with Remote Method Invocation;
    S. define and work with threads.

V. CONTENT:
    A. Language grammar and syntax:
        1. loops
        2. if conditions
        3. I/O capabilities
        4. file processing (sequential and random files)
        5. variables
        6. string processing
        7. display format
        8. arrays
        9. switch structure
    B. Creating queues, linked lists, stacks and trees with vector
    C. methods. Hash Tables
    D. Creating sorts in Java for numeric and string data
    E. (bubble, insert, shell, selection and quick)
    F. Binary search algorithm
    G. Projecting MIDI and WAV sound files over the web and within Swing.
    H. Setting up the java Multimedia Framework on your system
    I. Accessing sequential and random files within Swing and from an Applet
    J. Access databases and ODBC using Swing with Access databases.
    K. Overview of Remote Method Invocation
    L. Java Beans introduction
    M. Java Network capabilities
    N. Threads

VI. METHODS OF INSTRUCTION:
    A. **Lecture** -

VII. TYPICAL ASSIGNMENTS:

A. Using Vectors to Create Data Structures 1. Using a JOptionPane.showInputDialog to enter data 2. JOptionPane.showMessageDialog to show the output 3. Create a Linked List with the following capabilities: a. Add an item to the end of the list; b. Add an item to the beginning of the list; c. Ask the user where to place the item within the list; d. Remove the first item from the list; e. Remove the last item from the list; f. Remove an item anywhere in the list; g. Replace any existing item's value with another value; h. Display elements as the list changes (first to last and in reverse order, last to first); i. Check for an empty list; j. Exit. 4. You would probably wish to store your items in a vector with the following methods: a. Vectorname.size( ) for the number of elements in the vector b. Vectorname.firstElement( ) for the first element in the vector c. Vectorname.lastElement( ) for the last element in the vector d. Vectorname.insertElementAt(variable, integerindex); e. Vectorname.removeElementAt(variable, integerindex); 5. Enter a series of string values. 6. Display the list elements as you add and remove elements. 7. Display messages if you try to remove nonexistent elements or remove elements from an empty list. a. Hint: You might wish to use a switch structure to control the "looping" as the user makes selections. B. Write a program that asks a worker's age and years of service and determines whether that worker is eligible for retirement based on the following rule: 1. All employees are eligible to retire at 65. 2. All employees are eligible to retire after 35 years' service. 3. At 60, employees may retire with 25 years service. 4. At 55, employees may retire with 30 years' service. C. Write a program that reads a list of names and bowling scores from a text data file and produces a report. The report will be displayed on the screen and printed to a text output file for printing.

VIII. EVALUATION:
    A. **Methods**
    B. **Frequency**
        1. Frequency of evaluation
            a. Recommend 2 or 3 exams plus final examination
            b. Recommend programming assignment to cover each topic within course content. Contents can be combined.

2. Types of Exam Questions
    a. Write a method smallerOf() that returns the smaller of two given integer values.
    b. Using a conditional expression, construct a method sum() that returns the value according to Gauss's formula if the value of its parameter is positive, and zero otherwise. In algebraic form, Gauss's formula is: sum= (n(n+1))/2
    c. One method for finding the base-b representation of a positive integer given in base-10 notation is to divide the integer repeatedly by b until a quotient of zero results. The successive remainders are the digits from right to left of the base-b representation. Write a program to accept variouis integers and bases and display the digits of the base-b representation for each integer. You may assume that each base is in the range 2 through 10.

IX. TYPICAL TEXTS:

1. Deitel and Deitel *Java How to Program.* 3rd ed., Prentice Hall, 1999.
2. Y. Daniel Yang *Java Programming.* 2nd ed., QueE&T, 1999.
3. Douglas Bell and Mike Parr *Java, for Students.* 2nd ed., Prentice Hall, 1999.

X. OTHER MATERIALS REQUIRED OF STUDENTS:

A. Standard, current Java System Developer's Kit (SDK) installed on the institution's lab stations.
B. Current GUI Java compiler/editor such as Borland's JBuilder.
C. Current version of Microsoft Internet Explorer and Netscape Navigator to test applets.