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 2015

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

Applications programming using Java for students already familiar with the concepts of programming. Topics will include in Applets, GUI programming and design using Swing, presenting data files over the web, elementary data structures (queues, linked list, stacks) and vectors, searching (linear, binary), sorting algorithms, Database programming using JDBC (Java Data Base Connectivity), Remote Method Invocation (RMI), and Java Beans. The student shall also be exposed to and experience developing Java applications and applets in the Linux/Unix environment(s)

3.00 Units Lecture 1.00 Units Lab

Strongly Recommended
CS 1 - Computing Fundamentals I
with a minimum grade of C
and

CS 7 - Introduction to Computer Programming Concepts
with a minimum grade of C
or

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 this course, it is strongly recommended that the student should be able to:

A. CS1
1. Design, create and compile C++ programs within multiple development environments and operating systems, including the use of command-line tools in Unix/Linux.
2. Interpret and apply C++ control structures for sequencing, selection and iteration.
3. Interpret and implement programmer-defined functions in C++.
4. Create and interpret expressions involving arithmetic and logical operators;
5. Interpret and apply arrays and simple programmer-defined data structures in C++.
6. Modify and expand short programs that use standard conditional and iterative control structures and functions.
7. Choose appropriate conditional and iteration constructs for a given programming task.
8. Apply the techniques of structured (functional) decomposition to break a program into smaller pieces.
9. Analyze and explain the behavior of simple programs.
10. Describe, interpret and apply the mechanics of parameter passing.
11. Discuss and apply the concept of algorithms in problem-solving processes.
12. Judge the correctness and quality of algorithms, identifying necessary properties of good algorithms.
13. Describe and apply effective debugging strategies.
14. Identify properties of variables and apply different forms of variable binding, visibility, scoping, and lifetime management.
15. Explain, interpret and apply elements of syntax related variable types, including type-checking, abstraction, type incompatibility and type safety.
16. Design, implement, test, and debug programs using basic computation, simple I/O, standard conditional and iterative structures, and the definition of functions.

B. CS7
1. Design simple algorithms to solve a variety programming problems.
2. Design and implement programs of short to medium length, using standard elements of programming languages such as variables, input/output, control structures, functions/methods and arrays.
3. Describe the software development life-cycle.

4. Describe the principles of structured and object-oriented programming and be able to describe, design, implement, and test structured and object-oriented programs using currently accepted methodology.
5. Explain what an algorithm is and its importance in computer programming.
6. Analyze and investigate program behavior to effectively alter or debug existing code.
7. Design and implement specific program steps and components to achieve desired program behavior.
8. Design and organize elements of a program using a structured representation such as pseudocode and/or flowcharts.
9. Design and implement simple graphical and command line user interfaces implementing the students algorithms.

IV. MEASURABLE OBJECTIVES:

Upon completion of this course, the student should be able to:

A. Explain and apply basic principles of software engineering.
B. Design and implement both command line and graphical user interfaces using Java built-in classes.
C. Design and implement both built-in and complex Java data types and variables.
D. Understand and implement Java multi-dimensional arrays and vectors.
E. Design and implement multi-threaded Java applications.
F. Create java programs that implement standard data structures such as queues, linked lists, stacks, trees and hash tables.
G. Create Java programs that use one or more sorting algorithms (e.g., bubble, insertion, selection, shell, quick).
H. Create Java programs that use linear and binary search algorithms.
I. Design and implement programs accessing network resources and client/server protocols.
J. Write, compile, test and debug java programs and applets using both command line and integrated development environments;
K. Design and implement event-driven programs;
L. Design and implement Java classes including inheritance within a class hierarchy.
M. Design and implement Java constructors and other methods within a class.
N. Apply file input/output within the implementation of a program using both sequential and random data access methodologies.
O. Create Java programs using JDBC and Relational Database Management Systems (RDBMS) such as MySQL, DB2 and/or Oracle.

V. CONTENT:

A. Language syntax and semantics
    1. variables
    2. I/O capabilities
    3. if and switch statements
    4. loops
    5. string processing and formatting
    6. arrays
    7. classes
    8. inheritance
    9. polymorphism
B. Event-driven programming
C. Multi-threading
D. Implementation of data structures (e.g., queues, stacks, trees) in Java
E. Introduction to searching and sorting algorithms
F. Accessing sequential and random files
G. RDBMS and JDBC using CLI and/or Swing with MySQL, DB2 or Oracle.
H. Overview of Remote Method Invocation
I. Java Beans introduction
J. Java network capabilities
K. Cross operating system development and testing (Linux, Windows, OS X, etc.)

VI. METHODS OF INSTRUCTION:

A. **Lecture** -
B. **Lab** -
C. **Projects** -
D. **Classroom Activity** -

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.
D. Implement a relational database that defines a small company, departments, employees, and jobs and take your retirement program

which processed data from a text file and, instead, process the same data from the RDBMS implementation.

VIII. EVALUATION:

A. **Methods**

1. Other:
   a. Programming Assignments, evaluated on correctness, completeness, timely submission, documentation and structure.
   b. Written Homework
   c. Exams and/or Quizzes
   d. Final Examination

B. **Frequency**

1. Programming Assignment should be given approximately once per week.
2. Written homework should be weekly and ongoing, but may optionally be presented as a tool for students' self-evaluation (e.g., with answers provided, or feedback by an interactive tool) rather than included in instructor's overall evaluation of students.
3. At least one midterm exam plus several quizzes throughout the semester.
4. One comprehensive final examination at the end of the semester.

IX. TYPICAL TEXTS:

1. Deitel, . *Java How to Program.* 10th ed., Prentice Hall, 2014.
2. Horstmann, Cay, and Gary Cornell. *Core Java Volume I -- Fundamentals.* 9th ed., Prentice Hall, 2012.
3. Horstmann, Cay, and Gary Cornell. *Java Volume II -- Advanced Features.* 9th ed., Prentice Hall, 2012.
4. Eclipse Integrated Development Environment. Eclipse Foundation, (Luna+).
5. WindowsBuilder. Eclipse Foundation, (Most Current).
6. NetBeans. Oracle, (Most Current).

X. OTHER MATERIALS REQUIRED OF STUDENTS:

A. Standard, current Java System Developer’s Kit (SDK) installed on the institution’s lab stations.
B. Current IDE and GUI Java compiler/editor such Eclipse WindowBuilder or NetBeans.
C. Current version of Mozilla Firefox, Apple Safari and/or Microsoft Internet Explorer.