

# Syllabus - CIS 36B @ De Anza College

## Intermediate Problem Solving in Java

### Office hours

Location - F51e - Mon/Wed - 1pm to 2:50pm. You can also call me at 408 864 5566 during my office hours.

### Faculty Information

Sukhjot Singh phone: 408 864 5566 email: singhsukhjot@fhda.edu Office Location: F51e

**Requisites:** Students may receive credit for either Computer Information Systems (36A and 36B) or 35A, but not both.

**Pre-requisite:** Computer Information Systems 36A.

**Advisory:** English Writing 211 and Reading 211 (or Language Arts 211), or English as a Second Language 272 and 273; Mathematics 114 or equivalent.

**Hours:** Four hours lecture, one and one-half hours laboratory (66 hours total per quarter).

**Description:** A systematic approach to the design, construction and management of computer programs, emphasizing design, programming style, documentation, testing and debugging techniques. Strings, multi-dimensional arrays and Classes. References: their use in arrays, parameters and containment. Introduction to linked lists.

### Student Learning Outcome Statements (SLO)

- Student Learning Outcome: Read, analyze and explain intermediate level Java programs.
- Student Learning Outcome: Design solutions for intermediate level problems using appropriate design methodology incorporating intermediate programming constructs.
- Student Learning Outcome: Create algorithms, code, document, debug, and test intermediate level Java programs.

### Course Objectives

- A. Create programs which demonstrate knowledge of manipulating data in arrays of one or more dimensions.
- B. Use Java String and StringBuffer classes for manipulation of strings.
- C. Use Object-oriented programming concepts to design applications and computer programs.
- D. Create programs which use Inner Classes.
- E. Define and use the basic linked list operations: Traverse, Search, Insert, Delete. Design,

code, and test programs using linked lists.

F. Demonstrate usage of data structures in Java.

G. Demonstrate the basics of Exception Handling in Java.

H. Write programs to demonstrate the usage of File I/O API in Java.

## **V. Expanded Description: Content and Form**

A. Create programs which demonstrate knowledge of manipulating data in arrays of one or more dimensions.

1. One-Dimensional Arrays
  - a. Binary Search
  - b. Insertion Sort
2. Two-Dimensional Arrays
  - a. Declaration
  - b. Initialization
  - c. Access of individual elements
  - d. Use with functions
3. Arrays of more than two dimensions
  - a. Declaration
  - b. Initialization
  - c. Access of individual elements
  - d. Ragged Arrays
4. Test design and debugging strategies

B. Use Java String and StringBuffer classes for manipulation of strings.

1. String Class
  - a. Character testing and case conversion functions
  - b. Software engineering: Strings stored as arrays
  - c. String class member functions
  - d. Using references to pass Strings
2. StringBuffer Class
  - a. I/O with StringBuffer objects
  - b. Sorting with string Objects
  - c. StringBuffer class member functions

C. Use Object-oriented programming concepts to design applications and computer programs.

1. Object-oriented design
  - a. Encapsulation and information-hiding
  - b. Separation of behavior and implementation
  - c. Unified Modeling Language (UML)
2. Classes and Subclasses
  - a. Constructors
  - b. Fields

- c. Methods to alter Object behavior
  - d. Association
  - e. Inheritance
  - f. Encapsulation - Hiding fields
  - g. Use of overloaded functions
  - h. Class hierarchies
  - i. Overriding functions
  - j. Polymorphism
  - k. Internal representations of objects and method tables
  - l. Final Classes
3. Abstract Classes and Methods
- a. Definitions
  - b. Implementing abstract classes
  - c. Contract implementation
4. Interfaces
- a. Defining an Interface
  - b. Implementing an Interface
  - c. Using Interface as a datatype
  - d. Extending Interfaces
- D. Create programs which use Inner Classes.
- 1. Software engineering: abstract data types
  - 2. Declare an inner class
  - 3. Access fields in inner classes
  - 4. References to inner classes
  - 5. Nested inner classes
  - 6. Inner classes and functions
    - a. Inner class as parameters passed by reference
    - b. Return an inner class from a function
  - 7. Arrays of inner classes
- E. Define and use the basic linked list operations: Traverse, Search, Insert, Delete. Design, code, and test programs using linked lists.
- 1. Traversal
  - 2. Search
  - 3. Insertion
  - 4. Deletion
  - 5. Dynamic Allocation
- F. Demonstrate usage of data structures in Java.
- 1. Generic type in Java
  - 2. List and collection Interface including Iterator and ListIterator interface type.
  - 3. Stacks and Queues

4. Sets and Set Interface
5. Maps and Map Interface
6. Tree implementations
7. Hash Tables
8. Comparable interface

G. Demonstrate the basics of Exception Handling in Java.

1. try
2. throw
3. throws
4. finally

H. Write programs to demonstrate the usage of File I/O API in Java.

1. Definition of Stream, Buffer and Filters
2. Text Stream - FileReader, FileWriter, BufferedReader, BufferedWriter
3. Binary Streams - FileInputStream, FileOutputStream, BufferedInputStream, BufferedOutputStream
4. Filters - DataInputStream, DataOutputStream
5. ObjectStreams ObjectInputStream and ObjectOutputStream and Serialization

### **Grading System for this course**

For Letter Grade:

- Grade: A+ assigned with 97% or higher
- Grade: A assigned with 93% or higher
- Grade: A- assigned with 90% or higher
- Grade: B+ assigned with 87% or higher
- Grade: B assigned with 83% or higher
- Grade: B- assigned with 80% or higher
- Grade: C+ assigned with 77% or higher
- Grade: C assigned with 73% or higher
- Grade: D+ assigned with 70% or higher
- Grade: D assigned with 63% or higher
- Grade: D- assigned with 60% or higher
- Grade: F assigned with 0% or higher

For Pass/No Pass:

- Grade: Credit assigned with 70% or higher
- Grade: No Credit assigned with 0% or higher
- Incomplete

Audit

Withdrawal

### **Grading**

Labs - 50% of the grade

Midterm - 25% of the grade

Final - 25% of the grade

### **Assignments Due Dates:**

You will be assigned 8 to 10 assignments during the quarter. Assignment details can be found on the Assignment page. Midterm and Final dates can also be found on Assignment page

### **Methods of Evaluating Objectives**

- A. Evaluation of programming assignments for correctness, use of design principles, documentation, efficiency and teamwork.
- B. One or more examinations requiring some programming, concepts clarification and exhibiting mastery of programming principles.
- C. A final examination requiring concepts clarification and exhibiting mastery of programming principles.

### **Texts and Supporting References**

Introduction to Java Programming, Comprehensive (10th Edition) [Paperback] - Y. Daniel Liang (Author)  
ISBN-13: 978-0133761313 ISBN-10: 0133761312

[Link to Amazon.com](#)

Earlier editions of the same text are fine to use. As long as it is 6th edition or later.

Another great reference is [Thinking in Java](#) by Bruce Eckel.

### **Attendance**

You are responsible for completing all work assigned in this class in a timely fashion. You do not have to contact me with a reason of absence. You should be enrolled in the class at De Anza College for getting course access and to attend the class.

### **Withdrawing**

Once you are added to the class it is your responsibility to withdraw. I will not drop you from the class. The earned grade will be assigned at the end of the quarter.

### **Academic Dishonesty**

You are encouraged to discuss the ideas presented in the class. Copying or Cheating of work will result in zero grade for that assignment and may result in a failing grade. Basically I cannot tolerate cheating. You must work your solutions independently and all assignments and tests should be your own original work NO MAKEUP TESTS WILL BE GIVEN. You must pass the final to get a passing grade in this class.

### **Submitting Lab Assignments**

All assignments must be submitted electronically using the following guidelines.

Pl. email your assignments to **cislabs05@gmail.com**

Include the following information in the subject line

1. Your section #
2. Lab #
3. Your legal name (as it shows on academic records)

Use Text files for everything you submit.

You may submit files only with the following extension

- .txt (any design notes you want me to look at)
- .java (your source code.)
- .jpg or .gif (if you use any images for graphics programming)

You must include a readme.txt (for lab4 onwards) providing instructions to review and run your code.

Adequately test your code and run the test run of your code in a file called testrun.txt.

Every file should have the following information

- Your Name
- Class and Section
- Assignment Number
- Due Date
- Date Submitted

If you submit more than one file you must use winzip to compress all files into a single zip file and submit.

All Assignments are submitted by email to cislabs05@gmail.com.

Subject with each submission should be stated as - "CIS 36B - Lab <#>" - Replace # with the assignment number you are submitting.

## Lab Grading Criteria

- Full programming assignments will be evaluated with consideration given to
  - Accuracy (does the program solve the computing problem)
  - Adherence to Object Oriented Programming Methodology techniques (for Assignment 2 onwards)
  - Code readability and appearance
  - Naming Conventions
  - Documentation
  - Timeline
  - Professional Presentation

## Software

Download Java Standard Edition (latest version). Follow the installation instructions provided on the same page.

Mac users have java pre-installed and available in the Unix Shell on Mac OS. If you prefer a GUI based IDE then work with Eclipse.

Here is a video that might help - <http://www.youtube.com/watch?v=Otlva4ZHfqc>

## Lab Topics

- A. Array of References.

- B. Searching a one-dimensional array using binary search.
- C. Sorting a one-dimensional array using insertion/bubble sort.
- D. Two-dimensional arrays.
- E. Passing Object by references
- F. Inner Classes
- G. Strings.
- H. Design classes and subclasses
- I. Implementing Interfaces and Abstract Classes
- J. File IO
- K. Data Structures and Exception Handling