Indian Institute of Information Technology, Allahabad
Object Oriented Methodology (OOM) -2015
Lab Assignment-2

TAs: Bharat Singh, Nidhi Kushwaha, Monika Rani, Balasaheb

Date of assignment: 29/07/2015 **Due Date: 05/7/2015** **Instructor: Prof. O. P. Vyas & Dr. Ranjana Vyas**

Note: 1) All Assignments should be done independently.
2) Assignment should be evaluated within the deadline is essential.
3) Determine Class and object, .For each object, determine its attributes and behaviors.
4) Write simple code in java for printing appropriate value for class instances.

Q1. Suppose a company is setting up a software system to match up members of a car sharing pool. The company will provide the system to its own staff to use or to franchisees (which are just other companies which license the system and act as brokers). The software engineer engaged to develop the system needs to discuss the requirements with the company manager to establish exactly what they want. Suppose a segment of the initial discussion goes something like this:

Software Engineer (SE): "So you're saying that car sharers will be able to register by telephoning the office and speaking to someone there who will enter the details into the system."

Company Manager (CM): "Yes. Either the franchisee or more likely one of the office staff will take the call and enter the details into the computer."

SE: "Who are the office staff?"

CM: "Well there's one or two clerks, a receptionist and a supervisor. They all have a role in the administration of the system."

SE: "What will they be entering?"

CM: "Oh, the person's name and address, details of the journeys' they want to share, any preferences they have, such as being a non-smoker."

SE: "Is that the only way that this information will get into this system?"

CM: "No, it could also be transferred in from the national web-server."

SE: "How will this information be used?"

CM: "In two ways. Firstly it will be used to match up potential car sharers, and secondly, it will be used to produce a management report for the franchisee showing the number of registrations per week, whether they come from the web-server or by telephone and breaking them down by area."Based on this discussion, the Software Engineer agrees to draw up the preliminary class diagram, so that they can further define the system requirements at the next meeting.

Determine Class and object. For each object, determine its attributes and behaviors. Write simple code in java for printing appropriate value for identified class instances.

Q2. Suppose that a program is to be written in Java to solve the following problem: A video rental store wants a program to keep track of its movies. It rents VHS and DVD movies, with each movie given a unique inventory number. Each customer must have a phone number, which is used as his or her membership number. The program needs to keep track of every customer and every movie, including information such as whether a movie is rented or available, who has it rented, and when it is due back. Employees of the store receive a commission on sales of non-movie items such as candy and popcorn, so this information needs to be maintained as well.

1. Determine the objects in the problem domain
2. For each object, determine its attributes and behaviors.

Write a Java class for each of the classes.

1. When writing your classes, keep the following in mind:

Each class you write should be public; therefore, each class needs to appear in a separate source code file.

We have not discussed the details of writing methods. (Methods are discussed in the next chapter.) Focus on which methods each class should have, but don't worry about how they should be implemented. Within each method, use the System.out.println() method to display the name of the method. Be sure to save all your classes in the same directory on your hard drive

2. Writing a program to tie the classes together to make everything functional is not feasible without implementing all the methods. For now, just write and compile each class.

3. Write a program named VideoStore that creates an instance of each of your classes, initializes their fields, and invokes the methods to practice accessing the fields and methods of objects.

4. Compile and run your VideoStore program.

Determine Class and object. For each object, determine its attributes and behaviors. Write simple code in java for printing appropriate value for identified class instances.

Q3. Write a Java program that uses the **System.out.prinlnt()** method to display an email signature (name, title, email address, and so forth).

Perform the following steps:

Part-A

1. Create a new subdirectory in your javafiles directory, called Lab1_1.
2. Using your text editor, start with a public class called Signature.
   Add the main() method within the Signature class.
3. Within main(), use the System.out.println() method to print your name at the command prompt.
4. Similarly, print your title on a line, then your email address, Web site URL, or phone numbers. (Display any information you want in your signature. Just remember to use a semicolon after each println() statement.)

5. Save your Signature source code in the Lab1_1 folder in a file called Signature.java.
6. Compile Signature.java using the javac tool (or JAVA IDE).

Result seems to be like this:

```
Rich Raposa
Java Trainer
richraposa@javalicense.com
http://www.javalicense.com
```

Part-B

1. You have to input arguments that get passed to the main() method as strings.
2. There are three command-line arguments
   For Example :hello 27 Rich Raposa.

Hint: (Arguments are separated by spaces unless placed in double quotes.) These three arguments are passed into main() and placed in the args parameter. The args parameter is an array of strings that can hold as many command-line arguments as you enter.

3. Display these arguments using your source code. // **System.out.println("Name: " + args[2]);**

Hint: To access these arguments within main(), you use args with a subscript in square brackets. For example, args[0] is the first argument, args[1] is the second, and so on. In the current example, args[0] will be the string
"hello," args[1] will be "27," and args[2] will be "Rich Raposa."

.