

# ***Object Oriented***

## ***Course Project***

### **1. Introduction**

An object-based application is an application that involves objects created out of classes. These objects interact with each other to fulfill certain functionalities.

The focus of this project is to develop an application written in java which involves the use of object oriented programming. This project is a group project. As a student you will be given a chance to come up with an idea for your project. However, the project must satisfy the requirements given in this document.

This document shall be used as a set of guidelines. You are allowed to make necessary additions and/or changes to the requirements with prior approval from your instructor.

### **2. Requirements**

Propose and implement an application, that contributes to solving a real life problem. Your application should include at least the following:

#### **2.1 Classes**

3 to 4 classes each one of them should have:

- A. A number of private and public attributes. The private attributes (variables) will need corresponding set/get methods.
- B. 2 or more constructors, one of them should take all the class's data members (attributes) as arguments.
- C. Some functionality methods that can demonstrate the relationship between different classes. For example, assume that we have a system for a hotel, where a customer can book a room. Each customer has an ID (private) and a name. Each room has a number and price. The relationship between the Customer class and the Room class is **books** as illustrated in the following class diagram:

Customer		Room
- ID: String	books	+ No: int
+ name: String		+ regularPrice: double
+ printDetails(): void		

This relationship should be implemented in the Customer side as a method that takes a Room object as an argument, so when a customer books a room, the room object will be stored for example in an ArrayList or a file in case if the customer is allowed to book more than one room.

- D. Each class should have a printDetails method that will print out to the user the details of an object .
- E. One Main (Test) class to create objects out of the classes in your application, and to demonstrate the interaction between these objects.

## 2.2 Objects

- A. In the Main (Test) class create at least 3 objects out of each class using different constructors.
- B. After creating the objects, show how could you read or change the values of the private attributes.
- C. Demonstrate the interaction (relationships) between these created objects. For example, assuming that you have created a customer object (c1) and a room object (r1), you can use the method *books* to book r1 for c1 as follows:

c1.books(r1);

- D. Make a demo that shows the details to the user for each created object.

## 2.3 Files and Arrays

- A. When you create an object in the Test class, read the values of the object variables from the user (e.g., the value of an ID).
- B. After creating the objects store them temporally in an array (ArraList preferable). Then store them permanently in a file.
- C. When printing out the detail for an object read them from that file.

## 2.4 Input Validation

- A. Apply input validation where needed , for example , in the example given above, the price of a room should not be a negative value, or you cant have two customers with the same ID.

## **2.5 Inheritance (optional)**

Apply the inheritance in your application where possible.

## **3. Project Stages**

### **3.1 Stage 1**

A written proposal explaining the problem that you are solving, your application, how many classes in it, and how these classes will interact with each other. Use Diagrams where possible.

### **3.2 Stage 2**

Upon approval of your proposal, submit a class diagram showing the classes, their data members, methods, and relationships between the classes.

### **3.3 Stage 3**

The final submission of your code and output (Screenshots). (A recorded demo video will be a great idea)