**Core Java**

**Employee Id: \_\_\_\_\_\_\_\_\_\_\_\_\_ Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Duration: 3 Hours                                                                      Marks: 25**

**Instructions:**

1. ***First 10 minutes are provided to read & understand the test paper.***

***Last 10 minutes are provided to properly submit the test answer folder.***

***These 20 minutes are not counted in test duration***.

1. Create a workspace on the desktop. Create a java project with GID as name. Save all your files in this project only. Upload your Project folder and SHIFT + DEL from desktop.
2. Return the test paper to invigilator at the end of the test.
3. Referring Java Documentation is allowed.
4. Write your employee id in top left corner of the test paper
5. Use of Eclipse editor is allowed.
6. **Read all the section of paper carefully.**
7. **Marks will be only awarded to working functionalities.**
8. **Switch OFF your Cell Phone.**

**Q 1 [ Marks: 15 ]**

AnytimeServices is a provider of different Telecom related products to their Customers. They maintain the Orders placed by different Customers. One customer can raise multiple Orders.

* Design a **Order** class with following **private attributes**

orderID int

orderType String

orderDescription String

orderStatus String

* Create parameterized constructor having orderID, orderType and orderDescription as arguments
* **Getter** methods for all attributes.
* **Setter** Method for **orderStatus**
* Order status should be either **“Open” or “Closed”**
* **If** order status is neither **“Open”** nor **“Closed”** then throw the User defined Exception **InvalidSatusException**

Design a class **Customer** class

customerID int

customerName String

orderList Array list of type **Order**

* Create parameterized constructor having customerID , customerName as arguments
* Also define a method **public void addNewOrder(Order)** which will add an order placed by customer.
* The class will have method **public ArrayList getOrderList()** which will return all the orders placed by the customer.
* The class will have method **public void changeOrderStatus (int** orderID, **String** status **)** which will change the status of given orderID.
* Write the main method in the class TestOrder which reads the detail of a customer as input from the user. Also read 2 order details for that customer and add them to the arraylist.
* Ask user to enter order Id and Status, modify the status accordingly.
* Display the order details for that customer.

|  |  |  |
| --- | --- | --- |
| **Q 1 - A** | **Steps** | Marks |
| a) | Proper design of the class **Order** | 2 |
| b) | Proper design of the class **Customer** | 2 |
| c) | Dealing with Exception | 2 |
| d) | Proper working of addNewOrder() method | 2 |
| e) | Proper working of getOrderList() method | 2 |
| f) | Proper working of changeOrderStatus() method | 2 |
| g) | Reading inputs from the user and testing the functionalities | 3 |

**Q 2 [ Marks: 10 ]**

Write a class **Student** with the below attributes.

rollNo int

name String

major String

Provide getters and setters to initialize the attributes.

Write a class **StoreStudent** which contains the main method. Read the details of an employee as input from the user and create a Student instance.

Serialize the object into a file whose name is given as input through command line. Also deserialize the same to print the details.

|  |  |  |
| --- | --- | --- |
| **Q 2** | **Steps** | Marks |
| a) | Designing the Employee class properly | 2 |
| b) | Using command line argument | 1 |
| c) | Writing the object into a file | 3 |
| d) | Reading the object from the file | 3 |
| e) | Proper comments, coding conventions & indentation | 1 |