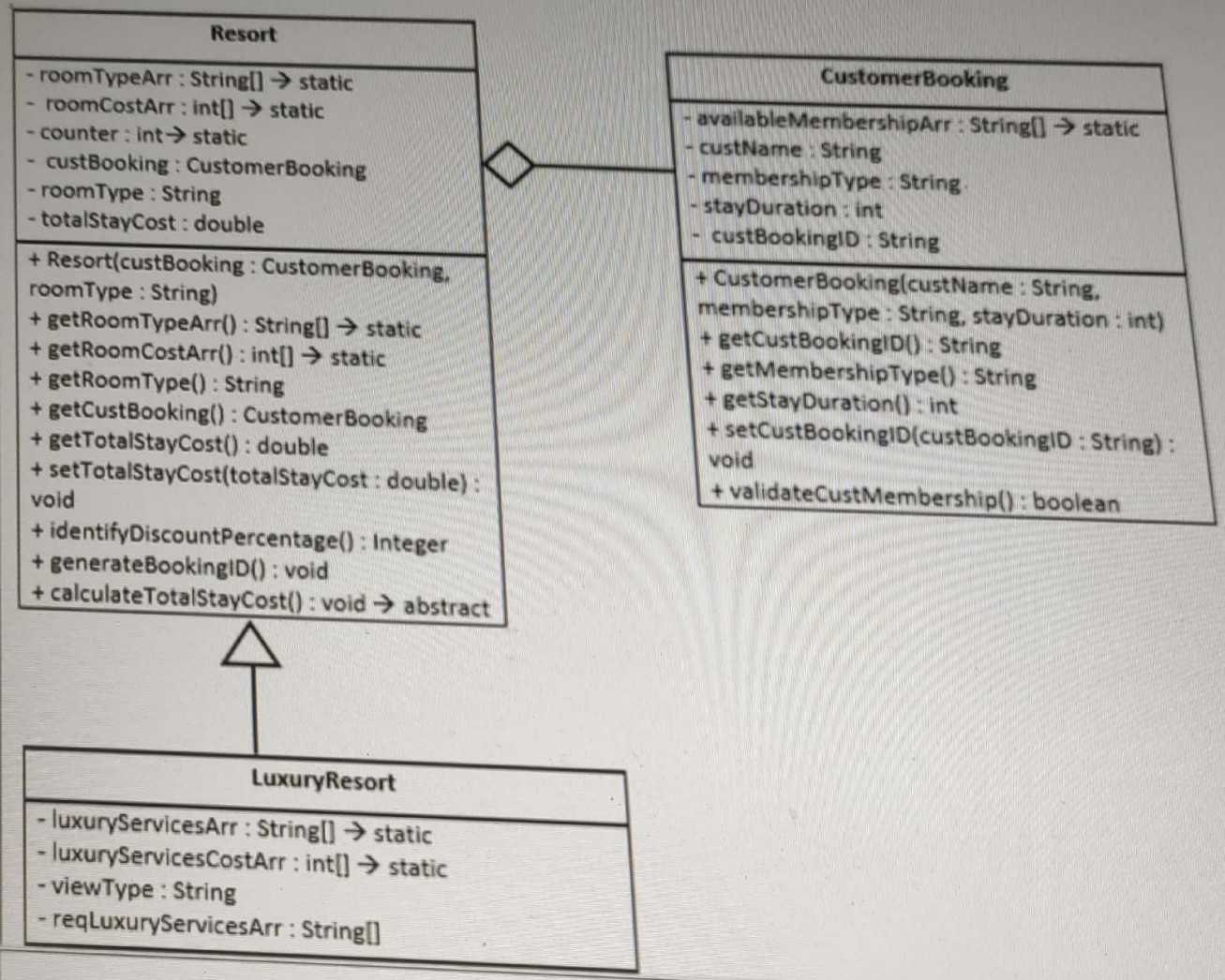
Question1: Object Oriented Programming: **[10 Marks]**

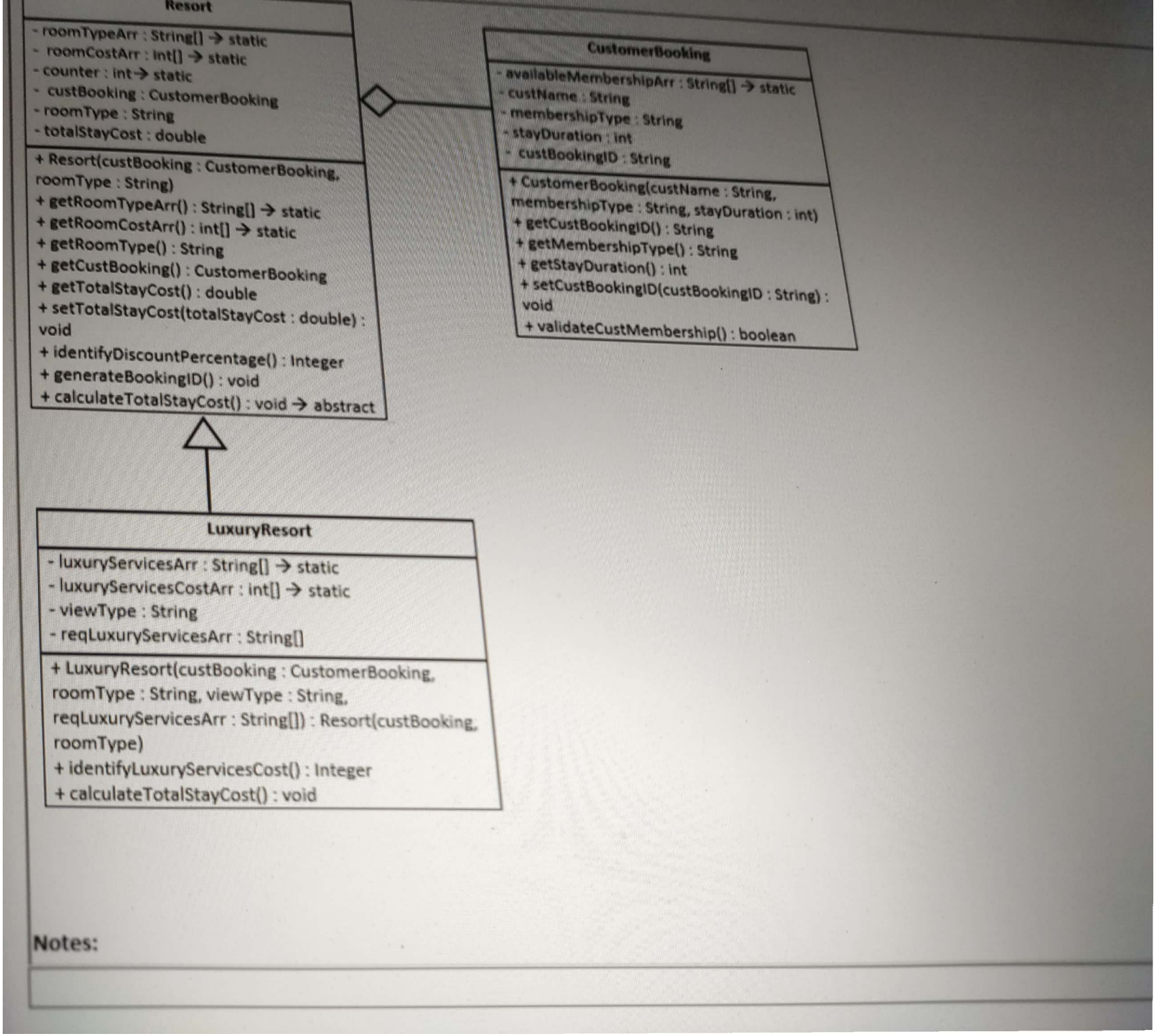
Description:

Paradise Stay, a popular Resort wants to automate its booking process based on various parameters.

Implement the class diagram given below to achieve the same.

Class Diagram:





- Do not include any extra instance/static variables and instance/static methods in the given classes
- **Case-insensitive** comparison is to be done if not explicitly mentioned
- Do not change any value or case of the given variables
- Read notes and examples for better understanding of the logic
- In the derived classes, the order of passing arguments to the constructor would be- base class variables followed by derived class variables

Implementation Details:

Class Name	Implementation Details
Resort	Partially implemented
LuxuryResort	Partially implemented
CustomerBooking	Fully implemented

CustomerBooking class:

availableMembershipArr:

- It is a static array (String[]) which has *availableMembership* (String) as its elements
- The initial values of the **availableMembershipArr** is as below:

availableMembershipArr	{"Premium", "Gold", "P

Note:

Type here to search

LuxuryResort	Partially implemented
CustomerBooking	Fully implemented

CustomerBooking class:

availableMembershipArr:

- It is a static array (String[]) which has *availableMembership* (String) as its elements
- The initial values of the **availableMembershipArr** is as below:

availableMembershipArr	{"Premium", "Gold", "Platinum"}

Note:

- This array is supplied and hence, no need to code
- Do not change the CASE of the elements in the array

Resort class:

roomTypeArr:

- It is a static array (String[]) which has *availableRoomType* (String) as its elements
- The initial values of the **roomTypeArr** is as below:

roomTypeArr	{"Regular", "Deluxe", "Executive"

- This array is supplied and hence, no need to code
- Do not change the CASE of the elements in the array

roomCostArr:

- It is a static array (int[]) which has *roomCostPerDay* (int) as its elements
- This array has one-to-one correspondence with the **roomTypeArr**
- The initial values of the **roomCostArr** is as below:

roomCostArr	{1500, 2500, 3500}

Note:

- This array is supplied and hence, no need to code

Example: If the **roomType** is "Deluxe", then the *roomCostPerDay* would be currency **2500**

identifyDiscountPercentage():

- This method identifies and returns the *discountPercentage* (Integer) based on the **membershipType** (String) of the customer and the **roomType** (Stri
- Refer the below table to identify the *discountPercentage*

membershipType	roomType	discountPercentage

Type here to search

identifyDiscountPercentage():

- This method identifies and returns the *discountPercentage* (Integer) based on the **membershipType** (String) of the customer and the **roomType** (String)
- Refer the below table to identify the *discountPercentage*

membershipType	roomType	discountPercentage
"Gold"	"Executive"	10
	"Deluxe"	8
"Platinum"	-	15
Other values	-	0

ote: Perform **case-sensitive** comparison for **membershipType** and **case-insensitive** comparison for **roomType**

ample: If the **membershipType** is "Gold", and **roomType** is "deluxe", then the *discountPercentage* would be **8**

nerateBookingID():

This method auto-generates and sets the **custBookingID** (String) of **CustomerBooking** class

The **custBookingID** must be prefixed by the first character of **roomType** in uppercase followed by the auto-generated value starting from 101

Subsequent **custBookingID** must be incremented by 1

Use static variable counter appropriately to implement the auto-generation logic

Type here to search

generateBookingID():

- This method auto-generates and sets the **custBookingID** (String) of **CustomerBooking** class
- The **custBookingID** must be prefixed by the first character of **roomType** in uppercase followed by the auto-generated value starting from 101
- Subsequent **custBookingID** must be incremented by 1
- Use static variable **counter** appropriately to implement the auto-generation logic

Example: The first **custBookingID** would be **"D101"** if the **roomType** is "deluxe", the second would be **"E102"** if the **roomType** is "Executive" and so on

LuxuryResort class:

luxuryServicesArr:

- It is a static array (String[]) which has *luxuryService* (String) as its elements
- The initial values of the **luxuryServicesArr** is as below:

luxuryServicesArr	{"Spa", "Pool", "Inroom-Dine"}

Note:

- This array is supplied and hence, no need to code

Note:

- This array is supplied and hence, no need to code
- Do not change the CASE of the elements in the array

luxuryServicesCostArr:

- It is a static array (int[]) which has *serviceCost* (int) as its elements
- This array has one-to-one correspondence with the **luxuryServicesArr**
- The initial values of the **luxuryServicesCostArr** is as below:

luxuryServicesCostArr	{800, 500, 400}

Note:

- This array is supplied and hence, no need to code

Example: If the *luxuryService* required is "Spa", then the corresponding *serviceCost* would be currency **800**

identifyLuxuryServicesCost():

- This method calculates and returns the *luxuryServicesCost* (Integer)
- If the **reqLuxuryServicesArr** (String[]) is **null**, then *luxuryServicesCost* would be zero (0)
- Otherwise, For each *service* in the **reqLuxuryServicesArr**,
  - If the *service* is one of the elements in the **luxuryServicesArr**, identify the corresponding *serviceCost* from the **luxuryServicesCostArr**

Assumption: Valid values for **reqLuxuryServicesArr** will be null or an array with String elements. Only valid values will be passed

Note:

- Perform **case-insensitive** comparison
- No need to validate the assumption

Example: If the **reqLuxuryServicesArr** is {"Spa", "Pool"}, then *luxuryServicesCost* would be currency **1300**

calculateTotalStayCost():

- This method sets the **custBookingID** and calculates and sets **totalStayCost** (double) based on the following logic:
- Invoke the **validateCustMembership()** method of **CustomerBooking** class
- If the above method returns **True** and the **stayDuration** is more than zero,
  - Check if the **roomType** (String) is present as an element in the **roomTypeArr**. If present, identify the corresponding roomCostPerDay from the **roomC**
  - Otherwise, *roomCostPerDay* would be -1
  - If the *roomCostPerDay* is not -1,
    - Identify the *discountPercentage* by invoking the **identifyDiscountPercentage()** method

Exam Data Submit Help

- Check if the **roomType** (String) is present as an element in the **roomTypeArr**. If present, identify the corresponding roomCostPerDay from the r
- Otherwise, *roomCostPerDay* would be **-1**
- If the *roomCostPerDay* is not **-1**,
  - Identify the *discountPercentage* by invoking the **identifyDiscountPercentage()** method
  - Identify the *luxuryServicesCost* by invoking the **identifyLuxuryServicesCost()** method
  - Add the above identified *luxuryServicesCost* to the *roomCostPerDay* to obtain *stayCost* (int)
  - If the **viewType** is "lake", add additional currency **500** to the *stayCost*
  - Identify the *totalStayCost* (int) by multiplying *stayCost* with **stayDuration**
  - Apply the *discountPercentage* on the *totalStayCost*
  - Set the **totalStayCost** (double) with the obtained *totalStayCost*
  - Invoke the **generateBookingID()** method
- Otherwise, set the **custBookingID** to "**NA**" and **totalStayCost** to **-1.0**

- Otherwise, set the **custBookingID** to "**NA**" and **totalStayCost** to **-1.0**

Assumptions:

- The valid values for **viewType** would be "lake" and "garden". Only valid values will be passed
- The **stayDuration** would be in number of days

Note:

- No need to validate the assumptions

Note:

- No need to validate the assumptions
- Perform **case-sensitive** comparison for **viewType** and **case-insensitive** comparison for **roomType**

Example:

If the **custName** is **"John"**, **membershipType** is **"Gold"**, **stayDuration** is **3** days, **roomType** is **"deluxe"**, **viewType** is **"lake"**, **reqLuxuryServicesArr** is **{"Spa", "Pool"}**, then the **custT**
totalStayCost would be currency **11868.0.**

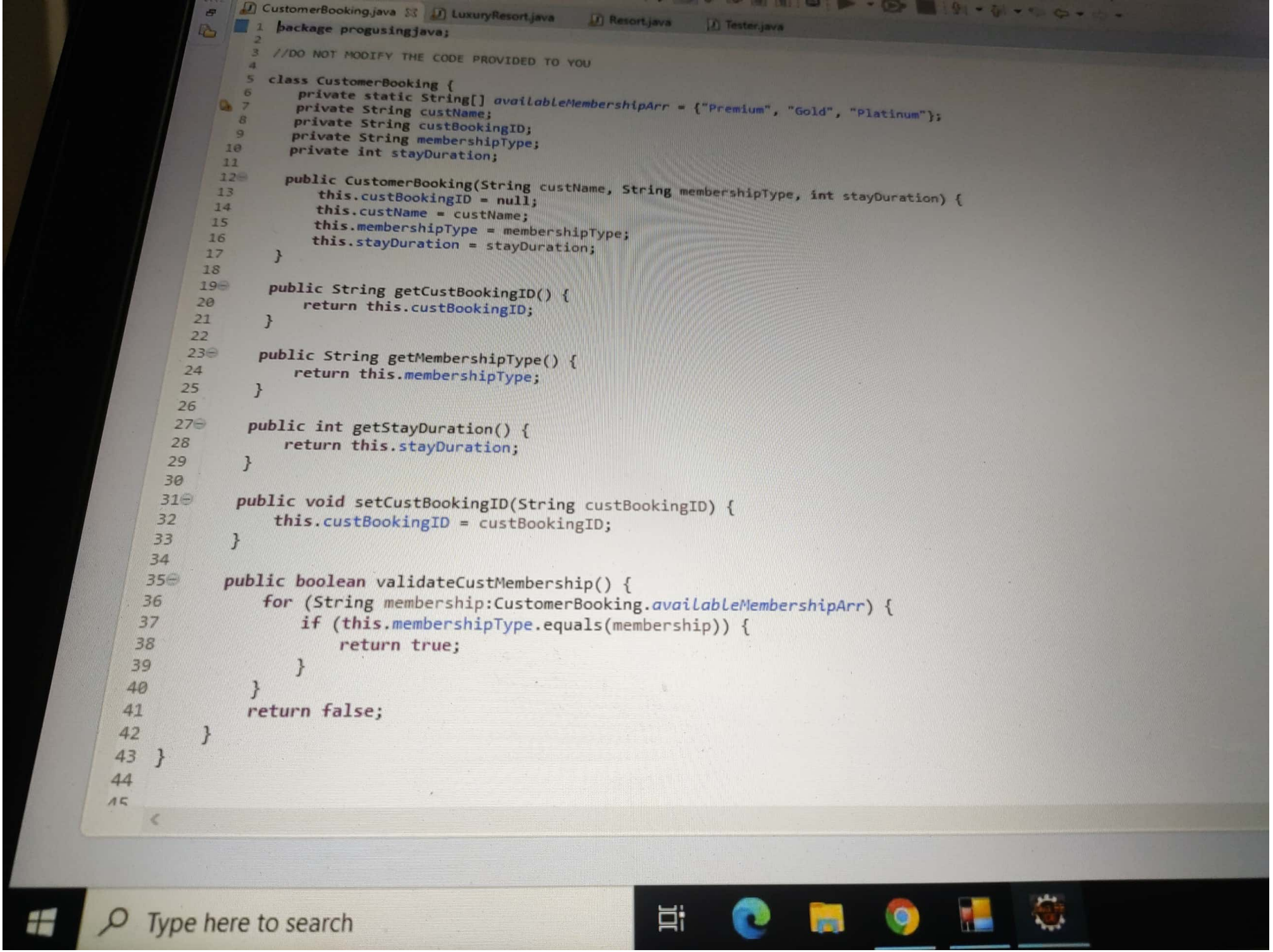
Question 2: Data Structures **[5 Marks]**

Problem Statement:

Description: Consider the non-empty **inStrStack** (String Stack) (Top→ Bottom) .

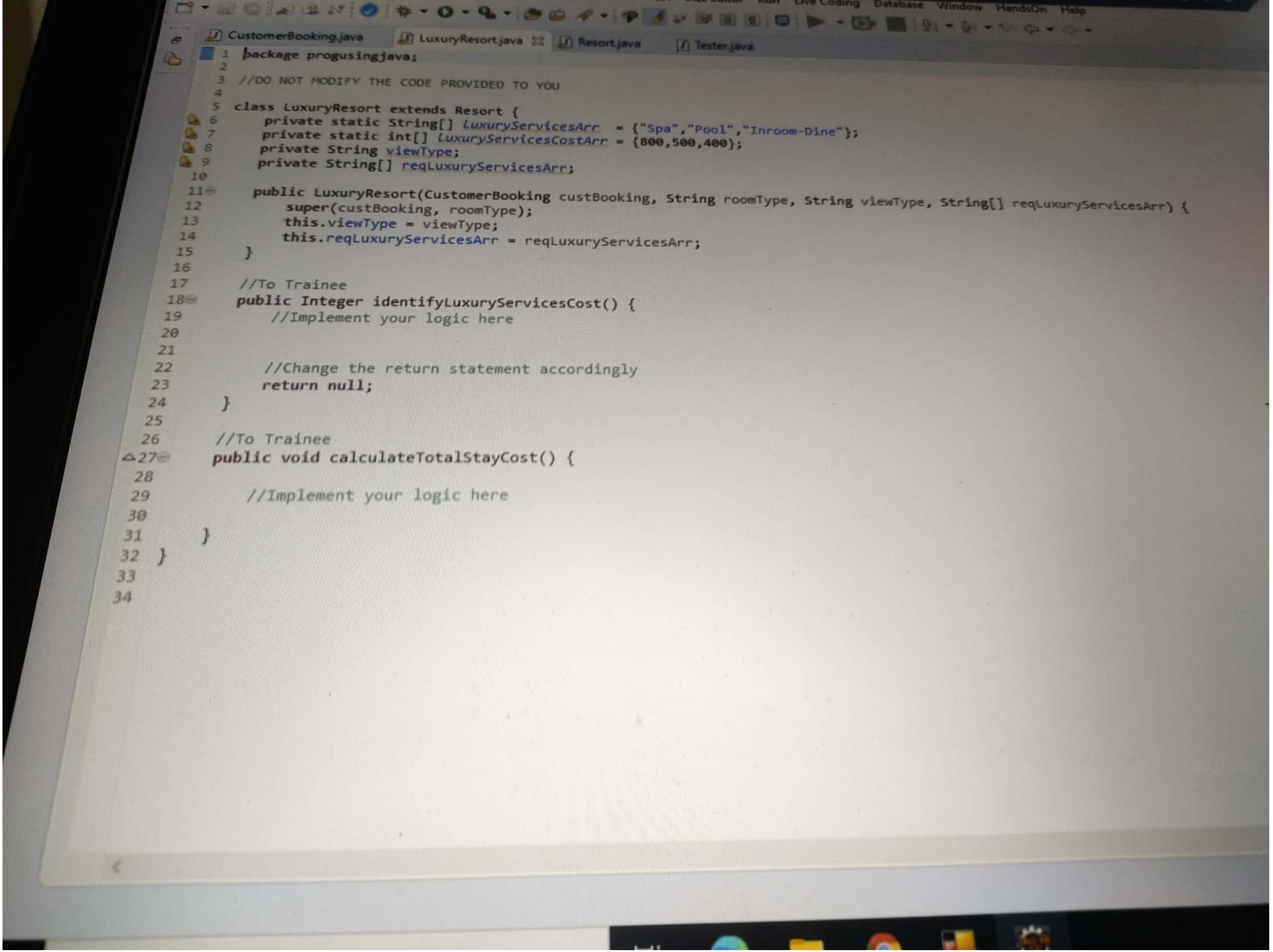
Note that each *element* of the **inStrStack** would contain the *itemDescription* followed by '-' followed by *distance* and *priority*. Here, *distance* is represented as 'C' for the orders
orders which are far to deliver and have a *priority* 1 or 2

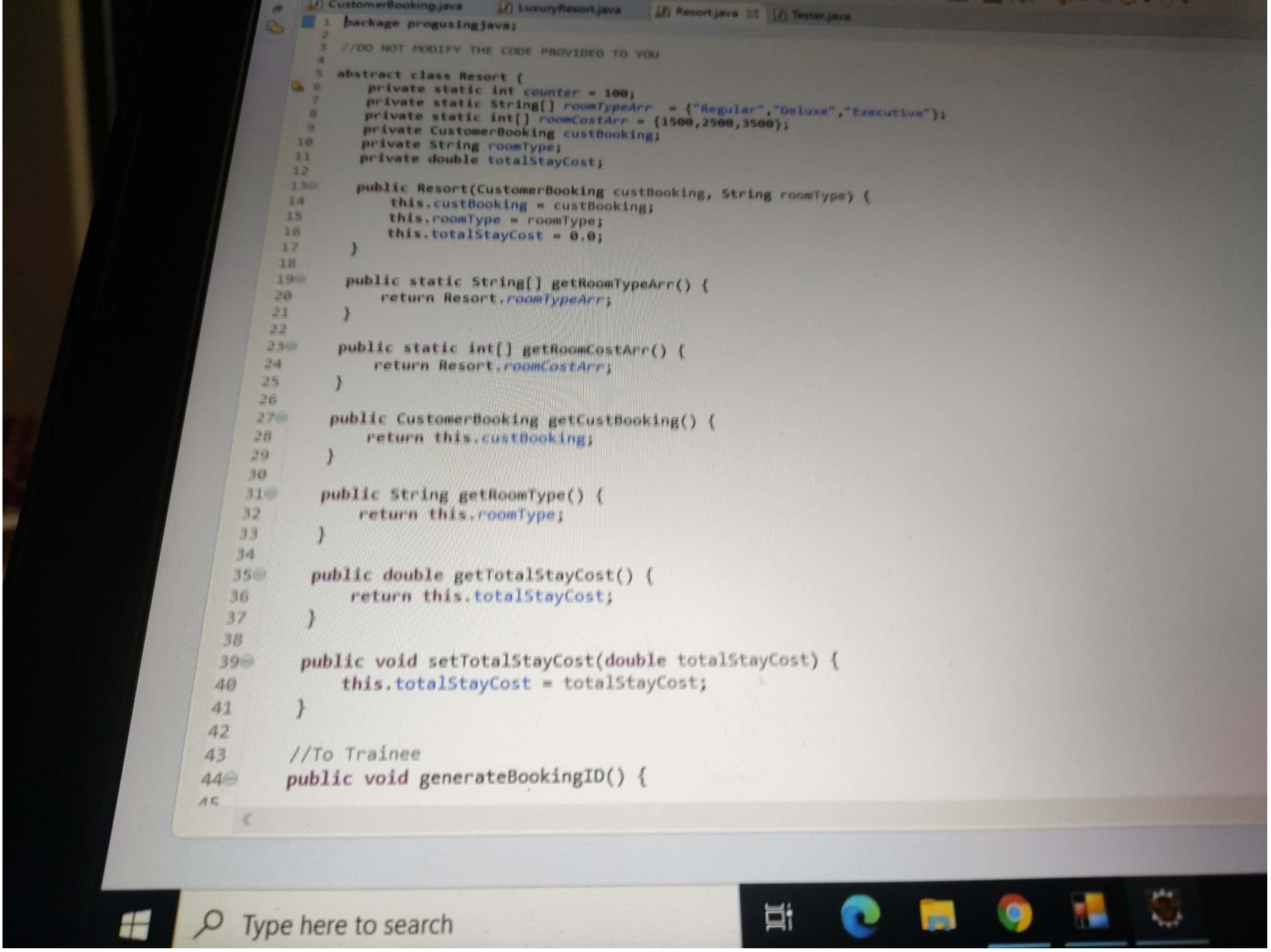
Write a Java program that accepts the above **inStrStack** as input parameter and returns the **outStrQueue** (String Queue) (Front → Rear) based on the below logic:



CustomerBooking.java | LuxuryResort.java | Resort.java | Tester.java

```java
package progusingjava;

//DO NOT MODIFY THE CODE PROVIDED TO YOU

class CustomerBooking {
    private static String[] availableMembershipArr = {"Premium", "Gold", "Platinum"};
    private String custName;
    private String custBookingID;
    private String membershipType;
    private int stayDuration;

    public CustomerBooking(String custName, String membershipType, int stayDuration) {
        this.custBookingID = null;
        this.custName = custName;
        this.membershipType = membershipType;
        this.stayDuration = stayDuration;
    }

    public String getCustBookingID() {
        return this.custBookingID;
    }

    public String getMembershipType() {
        return this.membershipType;
    }

    public int getStayDuration() {
        return this.stayDuration;
    }

    public void setCustBookingID(String custBookingID) {
        this.custBookingID = custBookingID;
    }

    public boolean validateCustMembership() {
        for (String membership:CustomerBooking.availableMembershipArr) {
            if (this.membershipType.equals(membership)) {
                return true;
            }
        }
        return false;
    }
}
```





```java
package progusingjava;

//DO NOT MODIFY THE CODE PROVIDED TO YOU

abstract class Resort {
    private static int counter = 100;
    private static String[] roomTypeArr  = {"Regular","Deluxe","Executive"};
    private static int[] roomCostArr = {1500,2500,3500};
    private CustomerBooking custBooking;
    private String roomType;
    private double totalStayCost;

    public Resort(CustomerBooking custBooking, String roomType) {
        this.custBooking = custBooking;
        this.roomType = roomType;
        this.totalStayCost = 0.0;
    }

    public static String[] getRoomTypeArr() {
        return Resort.roomTypeArr;
    }

    public static int[] getRoomCostArr() {
        return Resort.roomCostArr;
    }

    public CustomerBooking getCustBooking() {
        return this.custBooking;
    }

    public String getRoomType() {
        return this.roomType;
    }

    public double getTotalStayCost() {
        return this.totalStayCost;
    }

    public void setTotalStayCost(double totalStayCost) {
        this.totalStayCost = totalStayCost;
    }

    //To Trainee
    public void generateBookingID() {
```

```
    }

    public CustomerBooking getCustBooking() {
        return this.custBooking;
    }

    public String getRoomType() {
        return this.roomType;
    }

    public double getTotalStayCost() {
        return this.totalStayCost;
    }

    public void setTotalStayCost(double totalStayCost) {
        this.totalStayCost = totalStayCost;
    }

    //To Trainee
    public void generateBookingID() {

        //Implement your logic here

    }

    //To Trainee
    public Integer identifyDiscountPercentage() {
        //Implement your logic here


        //Change the return statement accordingly
        return null;
    }

    public abstract void calculateTotalStayCost();
}
```

CustomerBooking.java | LuxuryResort.java | Resort.java | Tester.java

```java
package progusingjava;

public class Tester {
    public static void main(String[] args) {
        CustomerBooking custOb = new CustomerBooking("John", "Gold", 3);
        String[] reqServices = {"Spa", "Pool"};
        LuxuryResort luxuryResortOb = new LuxuryResort(custOb, "deluxe", "lake", reqServices);
        luxuryResortOb.calculateTotalStayCost();
        System.out.println("Booking ID: "+custOb.getCustBookingID());
        System.out.println("Total Bill Amount: "+luxuryResortOb.getTotalStayCost());

    }
}

```