**Batch: A3 Roll No.: 16010122074**

**Experiment / assignment / tutorial No.06**

**Grade: AA / AB / BB / BC / CC / CD /DD**

**Signature of the Staff In-charge with date**

|  |
| --- |
| **TITLE: Case Study (for Class Diagram)** |

**AIM:** Draw class Diagram for the chosen Case Study. Clearly show

* + Attributes
  + Multiplicities between classes
  + Aggregations/compositions/Association between classes
  + Generalization between classes in the class diagram.

In addition, show the implementation of aggregation, association, composition and generalization between the classes.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Expected OUTCOME of Experiment:**

**CO1:** Understand the features of object oriented programming compared with procedural approach with C++ and Java.

**CO2**: Explore arrays, vectors, classes and objects in C++ and Java.

**CO3:** Implement scenarios using object oriented concepts (Drawing class diagram, relationship between classes, sequence diagram)

**CO4**: Explore the interface, exceptions, multithreading, packages

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Books/ Journals/ Websites referred:**

1. Ralph Bravado, Shai Simpson, “Java Programing From the Group Up” Tata McGraw-Hill.

2. Grady Brooch, Object Oriented Analysis and Design.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Pre Lab/ Prior Concepts:**

DefineClass, Methods, and Object.

Understanding of Aggregation, Association, Composition and Generalization between classes

**List of Classes:**

* **Hotel :**Contains details like customer details, book rooms, features, availability etc
* **Double Room :** Stores the details like name , contact number, gender of 2 different people.
* **Holder :**Stores the details of the types of rooms available
* **Single Room:** Stores the details like name, contact number, gender of a single person.
* **Food:** Gives details of different food items available.
* **Not available:** Determines that the room picked is not available for booking.

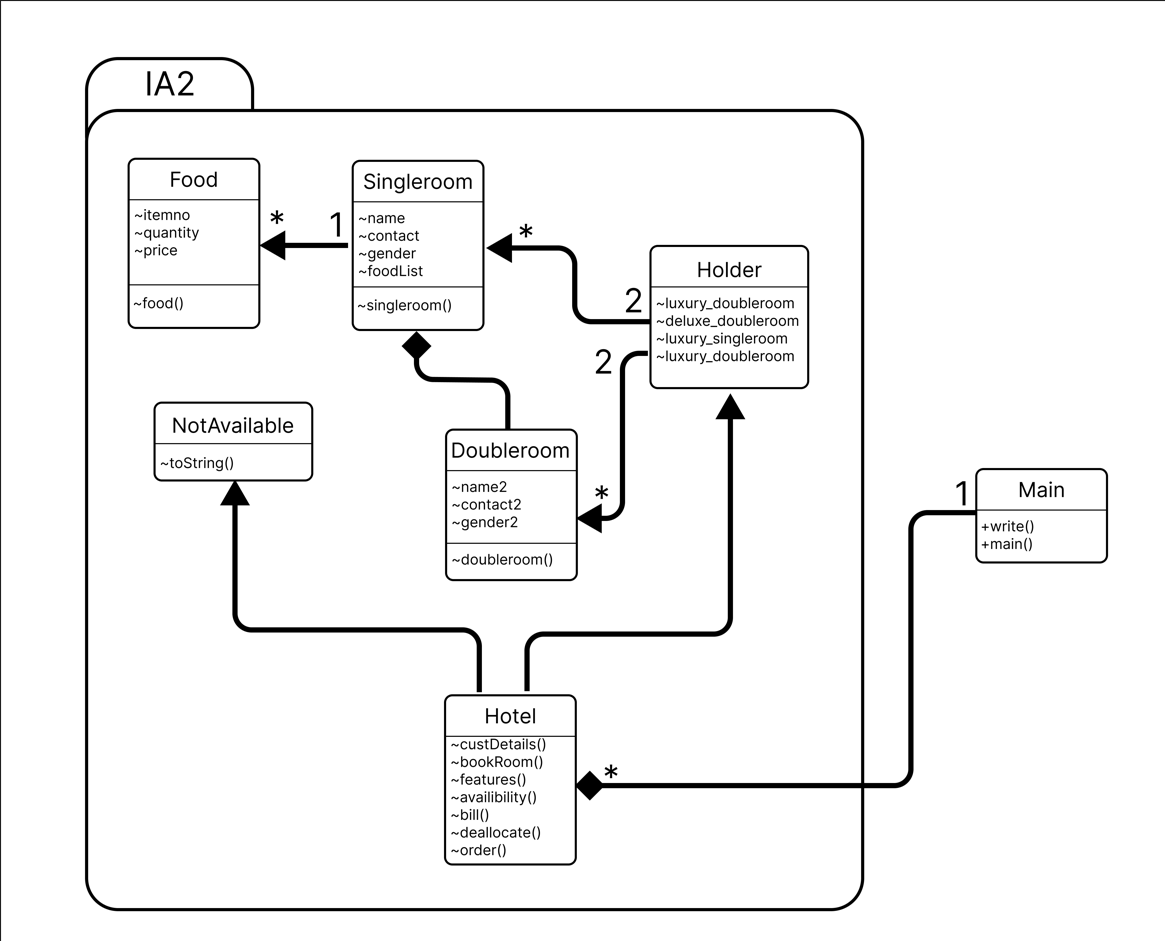
**Identify Attributes for each class:**

* **Hotel :**------
* **Double Room :** name2, contact2, gender2
* **Holder:** luxury\_doubleroom, deluxe\_doubleroom, luxury\_singleroom, deluxe\_doubleroom
* **Single Room:** name, contact, gender, foodList
* **Food:** itemno, quantity, price
* **Not available:** -----

**Identify List of Methods in each classes:**

* **Hotel :**custDetails() bookRoom() features() availability() bill() deallocate() and order()
* **Double Room :** doubleroom()
* **Holder :** ----
* **Single Room:** singleroom()
* **Food:** food()
* **Not available:** toString()

**Class Diagram:**



**Implementation details**: (Code)

**MAIN CLASS**

import java.io.\*;

import java.util.\*;

import ia2.\*;

class write implements Runnable

{

    Holder hotel\_ob;

    write(Holder hotel\_ob)

    {

        this.hotel\_ob=hotel\_ob;

    }

    @Override

    public void run()

    {

        try

        {

            FileOutputStream fout=new FileOutputStream("backup");

            ObjectOutputStream oos=new ObjectOutputStream(fout);

            oos.close();

            oos.writeObject(hotel\_ob);

        }

        catch(Exception e)

        {

            System.out.println("Error in writing "+e);

        }

    }

}

class Main

{

    public static void main(String[] args)

    {

        try

        {

            File f= new File("backup");

            if(f.exists())

            {

                FileInputStream fin=new FileInputStream(f);

                ObjectInputStream ois=new ObjectInputStream(fin);

                ois.close();

                Hotel.hotel\_ob=(Holder)ois.readObject();

            }

            Scanner sc = new Scanner(System.in);

            int ch,ch2;

            char wish;

            x:

            do

            {

                System.out.println("\nEnter your choice :\n1.Display room details\n2.Display room availability \n3.Book\n4.Order food\n5.Checkout\n6.Exit\n");

                ch=sc.nextInt();

                switch(ch)

                {

                    case 1:

                        System.out.println("\nChoose room type :\n1.Luxury Double Room\n2.Deluxe Double Room \n3.Luxury Single Room \n4.Deluxe Single Room\n");

                        ch2=sc.nextInt();

                        Hotel.features(ch2);

                        break;

                    case 2:

                        System.out.println("\nChoose room type :\n1.Luxury Double Room\n2.Deluxe Double Room \n3.Luxury Single Room \n4.Deluxe Single Room\n");

                        ch2=sc.nextInt();

                        Hotel.availability(ch2);

                        break;

                    case 3:

                        System.out.println("\nChoose room type :\n1.Luxury Double Room\n2.Deluxe Double Room \n3.Luxury Single Room \n4.Deluxe Single Room\n");

                        ch2=sc.nextInt();

                        Hotel.bookroom(ch2);

                        break;

                    case 4:

                        System.out.println("Room Number -");

                        ch2=sc.nextInt();

                        if(ch2>60)

                        System.out.println("Room doesn't exist");

                        else if (ch2>40)

                        Hotel.order(ch2-41,4);

                        else if (ch2>30)

                        Hotel.order(ch2-31,3);

                        else if (ch2>10)

                        Hotel.order(ch2-11,2);

                        else if (ch2>0)

                        Hotel.order(ch2-1,1);

                        else

                        System.out.println("Room doesn't exist");

                        break;

                        case 5:

                            System.out.println("Room Number-");

                            ch2=sc.nextInt();

                            if(ch2>60)

                            System.out.println("Room doesn't exist");

                            else if(ch2>40)

                            Hotel.deallocate(ch2-41,4);

                            else if(ch2>30)

                            Hotel.deallocate(ch2-31,3);

                            else if(ch2>10)

                            Hotel.deallocate(ch2-11,2);

                            else if(ch2>0)

                            Hotel.deallocate(ch2-1,1);

                            else

                            System.out.println("Room doesn't exist");

                            break;

                        case 6:

                            break x;

                }

                System.out.println("\nContinue : (y/n)");

                wish=sc.next().charAt(0);

                if(!(wish=='y'||wish=='Y'||wish=='n'||wish=='N'))

                {

                    System.out.println("Invalid Option");

                    System.out.println("\nContinue : (y/n)");

                    wish=sc.next().charAt(0);

                }

            }while(wish=='y'||wish=='Y');

            Thread t=new Thread(new write(Hotel.hotel\_ob));

            t.start();

        }

        catch(Exception e)

        {

            System.out.println("Not a valid input");

        }

    }

}

**SINGLEROOM CLASS:**

package ia2;

import java.util.ArrayList;

public class Singleroom

{

    public String name;

    public String contact;

    public String gender;

    public ArrayList<Food> food=new ArrayList<>();

    public Singleroom()

    {

        this.name="";

    }

    public Singleroom (String name,String contact,String gender)

    {

        this.name=name;

        this.contact=contact;

        this.gender=gender;

    }

}

**HOLDER CLASS:**

package ia2;

public class Holder

{

    public Doubleroom luxury\_doublerrom[]=new Doubleroom[10]; //Luxury

    public Doubleroom deluxe\_doublerrom[]=new Doubleroom[20]; //Deluxe

    public Singleroom luxury\_singleerrom[]=new Singleroom[10]; //Luxury

    public Singleroom deluxe\_singleerrom[]=new Singleroom[20]; //Deluxe

}

**NOT AVAILABLE CLASS**

package ia2;

public class NotAvailable extends Exception

{

    @Override

    public String toString()

    {

        return "Not Available !";

    }

}

**FOOD CLASS:**

package ia2;

public class Food{

    public int itemno;

    public int quantity;

    public float price;

    public Food(int itemno,int quantity)

    {

        this.itemno=itemno;

        this.quantity=quantity;

        switch(itemno)

        {

            case 1:price=quantity\*50;

            break;

            case 2:price=quantity\*60;

            break;

            case 3:price=quantity\*70;

            break;

            case 4:price=quantity\*30;

            break;

        }

    }

}

**DOUBLE ROOM CLASS**

package ia2;

public class Doubleroom extends Singleroom

{

    public String name2;

    public String contact2;

    public String gender2;

    public Doubleroom()

    {

        this.name="";

        this.name2="";

    }

    public Doubleroom(String name,String contact,String gender,String name2,String contact2,String gender2)

    {

        this.name=name;

        this.contact=contact;

        this.gender=gender;

        this.name2=name2;

        this.contact2=contact2;

        this.gender2=gender2;

    }

}

**HOTEL CLASS:**

package ia2;

import java.util.\*;

public class Hotel

{

    public static Holder hotel\_ob=new Holder();

    public static Scanner sc = new Scanner(System.in);

    public static void CustDetails(int i,int rn)

    {

        String name, contact, gender;

        String name2 = null, contact2 = null;

        String gender2="";

        System.out.print("InEnter customer name: ");

        name=sc.next();

        System.out.print("Enter contact number: ");

        contact=sc.next();

        System.out.print("Enter gender: ");

        gender=sc.next();

        if(i<3)

        {

            System.out.print("Enter second customer name: ");

            name2=sc.next();System.out.print("Enter contact number: ");

            contact2=sc.next();

            System.out.print("Enter gender: ");

            gender2 =sc.next();

        }

        switch (i)

        {

            case 1:hotel\_ob.luxury\_doublerrom[rn]=new Doubleroom(name, contact, gender, name2,contact2, gender2);

            break;

            case 2:hotel\_ob.deluxe\_doublerrom[rn]=new Doubleroom(name, contact,gender,name2,contact2,gender2);

            break;

            case 3:hotel\_ob. luxury\_singleerrom[rn]=new Singleroom(name, contact,gender);

            break;

            case 4:hotel\_ob.deluxe\_singleerrom[rn]=new Singleroom(name, contact,gender);

            break;

            default:System.out.println("Wrong option");

            break;

        }

    }

    public static void bookroom(int i)

    {

        int j;

        int rn;

        System.out.println("\n Choose room number from : ");

        switch (i)

        {

            case 1:

                for(j=0;j<hotel\_ob.luxury\_doublerrom.length;j++)

                {

                    if(hotel\_ob.luxury\_doublerrom[j]==null)

                    {

                        System.out.print(j+1+",");

                    }

                }

                System.out.print("/nEnter room number: ");

                try

                {

                    rn=sc.nextInt();

                    rn--;

                    if(hotel\_ob.luxury\_doublerrom[rn]!=null)

                    throw new NotAvailable();

                    CustDetails(i,rn);

                }

                catch(Exception e)

                {

                    System.out.println("Invalid Option");

                    return;

                }

                break;

            case 2:

                for(j=0;j<hotel\_ob.deluxe\_doublerrom.length;j++)

                {

                    if(hotel\_ob.deluxe\_doublerrom[j]==null)

                    {

                        System.out.print(j+11+",");

                    }

                }

                System.out.print("InEnter room number: ");

                try

                {

                    rn=sc.nextInt();

                    rn=rn-11;

                    if(hotel\_ob.deluxe\_doublerrom[rn]!=null)

                    throw new NotAvailable();

                    CustDetails(i,rn);

                }

                catch(Exception e)

                {

                    System.out.println("Invalid Option");

                    return;

                }

                break;

            case 3:

                for(j=0;j<hotel\_ob.luxury\_singleerrom.length;j++)

                {

                    if(hotel\_ob. luxury\_singleerrom[j]==null)

                    {

                        System.out.print(j+31+",");

                    }

                }

                System.out.print("InEnter room number:");

                try

                {

                    rn=sc.nextInt();

                    rn=rn-31;

                    if(hotel\_ob.luxury\_singleerrom[rn]!=null)

                    throw new NotAvailable();

                    CustDetails(i,rn);

                }

                catch(Exception e)

                {

                    System.out.println("Invalid Option");

                    return;

                }

                break;

            case 4:

                for(j=0;j<hotel\_ob.deluxe\_singleerrom.length;j++)

                {

                    if(hotel\_ob.deluxe\_singleerrom[j]==null)

                    {

                        System.out.print(j+41+",");

                    }

                }

                System.out.print("InEnter room number: ");

                try

                {

                    rn=sc.nextInt();

                    rn=rn-41;

                    if(hotel\_ob.deluxe\_singleerrom[rn]!=null)

                    throw new NotAvailable();

                    CustDetails(i,rn);

                }

                catch(Exception e)

                {

                    System.out.println("Invalid Option");

                    return;

                }

                break;

            default:

                System.out.println("Enter valid option");

                break;

        }

            System.out.println("Room Booked");

    }

    public static void features(int i)

    {

         switch (i)

        {

            case 1:

                System.out.println("Number of double beds : 1 \nAC : Yes\nFree breakfast: Yes\nCharge per day 4000 ");

                break;

            case 2:

                System.out.println("Number of double beds : 1 \nAC : No\nFree breakfast: Yes\nCharge per day 3000 ");

                break;

            case 3:

                System.out.println("Number of single beds : 1 \nAC : Yes\nFree breakfast: Yes\nCharge per day 2200 ");

                break;

            case 4:

                System.out.println("Number of single beds : 1 \nAC : No\nFree breakfast: Yes\nCharge per day 1200 ");

                break;

            default:

                System.out.println("Enter valid option");

                break;

        }

    }

    public static void availability(int i)

    {

        int j,count=0;

        switch (1)

        {

            case 1:

                for(j=0;j<10;j++)

                {

                    if(hotel\_ob.luxury\_doublerrom[j]==null)

                    count++;

                }

                break;

            case 2:

                for(j=0;j<hotel\_ob.deluxe\_doublerrom.length;j++)

                {

                    if(hotel\_ob.deluxe\_doublerrom[j]==null)

                    count++;

                }

                break;

            case 3:

                for(j=0;j<hotel\_ob.luxury\_singleerrom.length;j++)

                {

                    if(hotel\_ob.luxury\_singleerrom[j]==null)

                    count++;

                }

                break;

                case 4:

                for(j=0;j<hotel\_ob.deluxe\_singleerrom.length;j++)

                {

                    if(hotel\_ob.deluxe\_singleerrom[j]==null)

                    count++;

                }

                break;

            default:

                System.out.println("Enter valld option");

                break;

        }

        System.out.println("Number of rooms available : "+count);

    }

    public static void bill(int rn,int rtype)

    {

        double amount=0;

        String list[]={"Sandwich","Pasta","Noodles","Coke"};

        System.out.println("\n\*\*\*\*\*\*\*");

        System.out.println("Bill:-");

        System.out.println("\*\*\*\*\*\*\*");

        switch(rtype)

        {

            case 1:

                amount+=4000;

                System.out.println("\nRoom Charge - "+4000);

                System.out.println("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

                System.out.println("Food Charges:-");

                System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

                System.out.println("Item Quantity Price");

                System.out.println(".........................");

                for(Food obb:hotel\_ob.luxury\_doublerrom[rn].food)

                {

                    amount+=obb.price;

                    System.out.printf(list[obb.itemno-1],obb.quantity,obb.price);

                }

                break;

            case 2:

                amount+=3000;

                System.out.println("Room Charge- "+3000);

                System.out.println("\nFood Charges:- ");

                System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

                System.out.println("Item Quantity Price");

                System.out.println(".........................");

                for(Food obb:hotel\_ob.deluxe\_doublerrom[rn].food)

                {

                    amount+=obb.price;

                    System.out.printf(list[obb.itemno-1],obb.quantity,obb.price );

                }

                break;

            case 3:

                amount+=2200;

                System.out.println("Room Charge- "+2200);

                System.out.println("\nFood Charges:- ");

                System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

                System.out.println("Item Quantity Price");

                System.out.println(".........................");

                System.out.println("Item Quantity Price");

                System.out.println("------------------- ");

                for(Food obb:hotel\_ob.luxury\_singleerrom[rn].food)

                {

                    amount+=obb.price;

                    System.out.printf(list[obb.itemno-1],obb.quantity,obb.price);

                }

                break;

            case 4:

                amount+=1200;

                System.out.println("Room Charge- "+1200);

                System.out.println("\nFood Charges:- ");

                System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

                System.out.println("Item Quantity Price");

                System.out.println(".........................");

                for(Food obb: hotel\_ob.deluxe\_singleerrom[rn].food)

                {

                    amount+=obb.price;

                    System.out.printf(list[obb.itemno-1],obb.quantity,obb.price );

                }

                break;

            default:

                System.out.println("Not valid");

        }

        System.out.println("InTotal Amount- "+amount);

    }

    public static void deallocate(int rn,int rtype)

    {

        char w;

        switch (rtype)

        {

            case 1:

                if(hotel\_ob.luxury\_doublerrom[rn]!=null)

                System.out.println("Room used by "+hotel\_ob.luxury\_doublerrom[rn].name);

                else

                {

                    System.out.println("Empty Aready");

                    return;

                }

                System.out.println("Do you want to checkout ?(y/n)");

                w=sc.next().charAt(0);

                if(w=='y'||w=='Y')

                {

                    bill(rn,rtype);

                    hotel\_ob.luxury\_doublerrom[rn]=null;

                    System.out.println("Deallocated succesfully");

                }

                break;

            case 2:

                if(hotel\_ob.deluxe\_doublerrom[rn]!=null)

                System.out.println("Room used by "+hotel\_ob.deluxe\_doublerrom[rn].name);

                else

                {

                    System.out.println("Empty Already");

                    return;

                }

                System.out.println(" Do you want to checkout ?(y/n)");

                w=sc.next().charAt(0);

                if(w=='y'||w=='Y')

                {

                    bill(rn,rtype);

                    hotel\_ob.deluxe\_doublerrom[rn]=null;

                    System.out.println("Deallocated succesfully");

                }

                break;

            case 3:

                if(hotel\_ob.luxury\_singleerrom[rn]!=null)

                System.out.println("Room used by "+hotel\_ob.luxury\_singleerrom[rn].name);

                else

                {

                    System.out.println("Empty Already");

                    return;

                }

                System.out.println(" Do you want to checkout ? (y/n)");

                w=sc.next().charAt(0);

                if(w=='y'||w=='Y')

                {

                    bill(rn,rtype);

                    hotel\_ob.luxury\_singleerrom[rn]=null;

                    System.out.println("Deallocated succesfully");

                }

                break;

            case 4:

                if(hotel\_ob.deluxe\_singleerrom[rn]!=null)

                System.out.println("Room used by "+hotel\_ob.deluxe\_singleerrom[rn].name);

                else

                {

                    System.out.println("Empty Already");

                    return;

                }

                System.out.println(" Do you want to checkout ? (y/n)");

                w=sc.next().charAt(0);

                if(w=='y'||w=='Y')

                {

                    bill(rn,rtype);

                    hotel\_ob.deluxe\_singleerrom[rn]=null;

                    System.out.println("Deallocated succesfully");

                }

                break;

            default:

                System.out.println("\nEnter valid option : ");

                break;

        }

    }

    public static void order(int rn,int rtype)

    {

        int i,q;

        char wish;

        try

        {

            System.out.println("\n=======\n Menu: \n=========\n\n1.Sandwich\tRs.50\n2.Pasta\t\tRs.60\n3.Noodles\tRs.70\n4.Coke\t\tRs.30\n");

            do

            {

                i=sc.nextInt();

                System.out.print("Quantity- ");

                q=sc.nextInt();

                switch(rtype)

                {

                    case 1:

                        hotel\_ob.luxury\_doublerrom[rn].food.add(new Food(i,q));

                        break;

                    case 2:

                        hotel\_ob.deluxe\_doublerrom[rn].food.add(new Food(i,q));

                        break;

                    case 3:

                        hotel\_ob.luxury\_singleerrom[rn].food.add(new Food(1,q));

                        break;

                    case 4:

                        hotel\_ob.deluxe\_singleerrom[rn].food.add(new Food(i,q));

                        break;

                }

                System.out.println("Do you want to order anything else?  (y/n)");

                wish=sc.next().charAt(0);

            }while(wish=='y'||wish=='Y');

        }

        catch(NullPointerException e)

        {

            System.out.println("InRoom not booked");

        }

        catch(Exception e)

        {

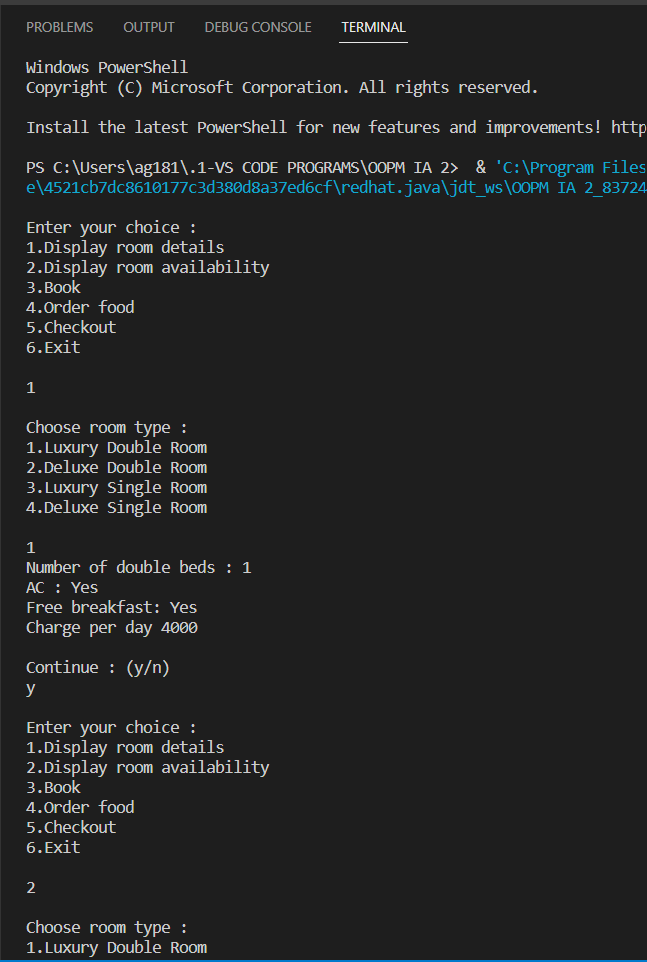
            System.out.println("Cannot be done");

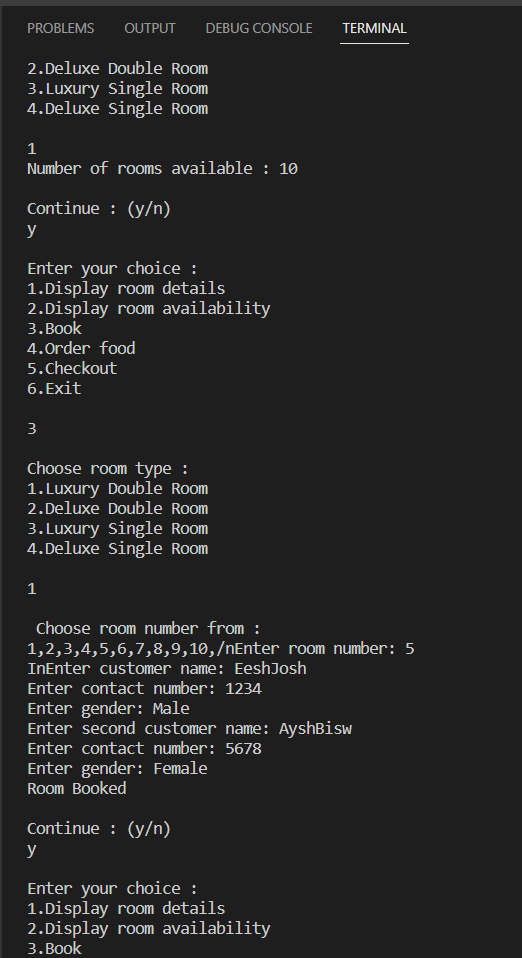
        }

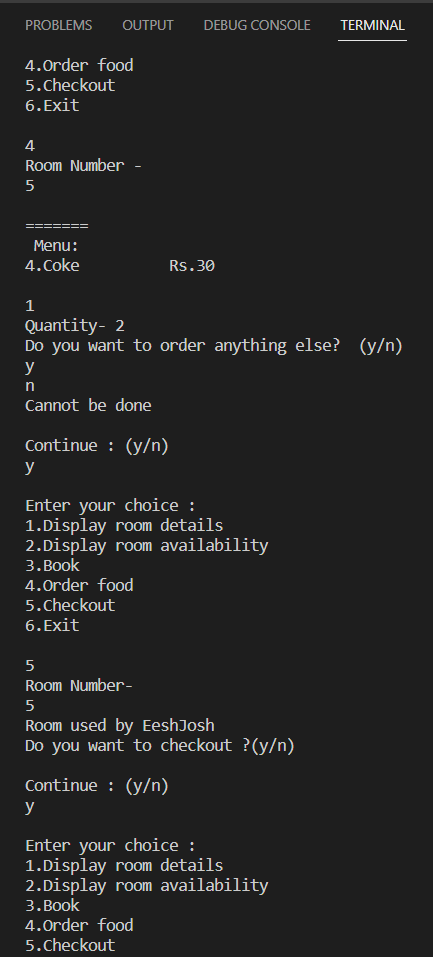
    }

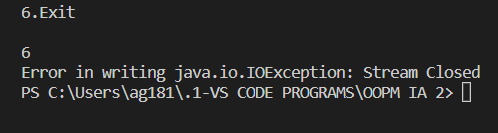
}

**OUTPUT:**

****







**Conclusion:** Class diagram successfully drawn for the chosen case and code implemented correctly.

**Date: \_\_\_\_\_\_\_\_\_\_ Signature of faculty in-charge**

**Post Lab Descriptive Questions**

**1.** **Consider the following class:**

public class TypeOfVariable{

public static int a;

int b,c;

public void printValue(){

int x = 10;

}

public static void main(String args[]){

TypeOfVariable object=new TypeOfVariable();

object.printValue();

}

}

a). What are the class/static variables? TypeOfVariable

b). What are the instance variables? Int a,b,c

c.)What are local variables? Int x

**2.What is the output from the following code:**

public class Test

{

    static int x = 11;

    private int y = 33;

    public void method1(int x)

    {

        Test t = new Test();

        this.x = 22;

        y = 44;

        System.out.println("Test.x: " + Test.x);

        System.out.println("t.x: " + t.x);

        System.out.println("t.y: " + t.y);

        System.out.println("y: " + y);

    }

    public static void main(String args[])

    {

        Test t = new Test();

        t.method1(5);

    }

}

**Output:**

Test.x: 22

t.x: 22

t.y: 33

y: 44