

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Programme** | **:** | **B,Tech(CSE)** | **Semester** | **:** | **Winter 2020-21** |
| **Course** | **:** | **Java Programming** | **Code** | **:** | **CSE1007** |
| **Faculty** | **:** | **Rajarajeshwari S** | **Slot** | **:** | **A1** |
| **Name** | **:** | **Prakriti Sharma** | **Reg. No.** | **:** | **19BCE1655** |

LINK TO THE VIDEO :

<https://drive.google.com/file/d/1hLcKDH_ofCvYwUVm7EHUq53he8ZLhSKU/view?usp=sharing>

**THE DOCUMENT HAS :**

**1.ANSWER TO THE CLASS DEFINITION PART.**

**2.SCREENSHOTS OF TEST CASES.**

**3.SCREENSHOT OF OUTPUT.**

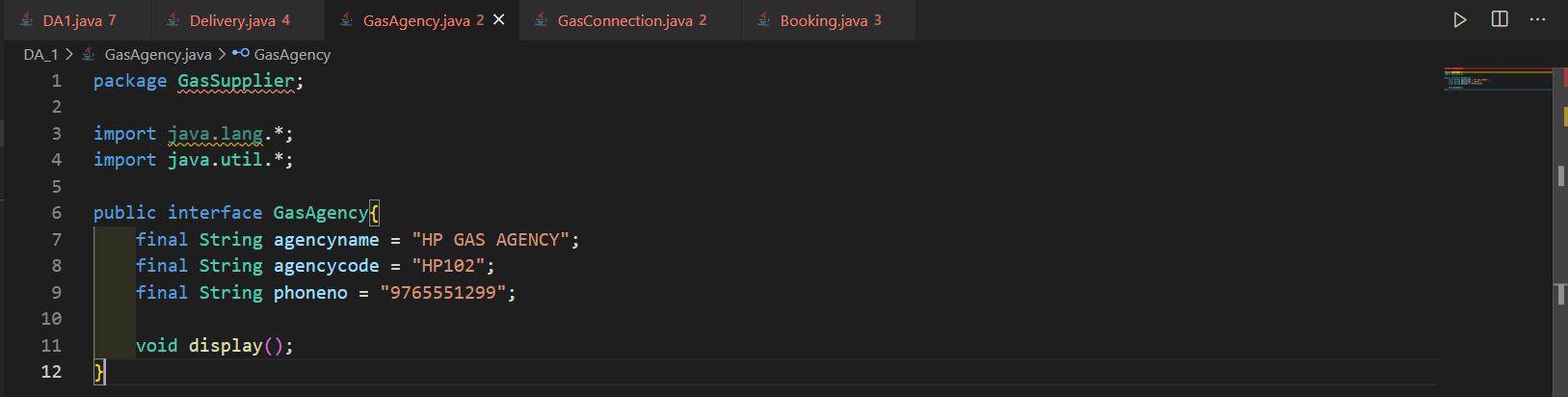
**4.PRE-DEFINED DATA**

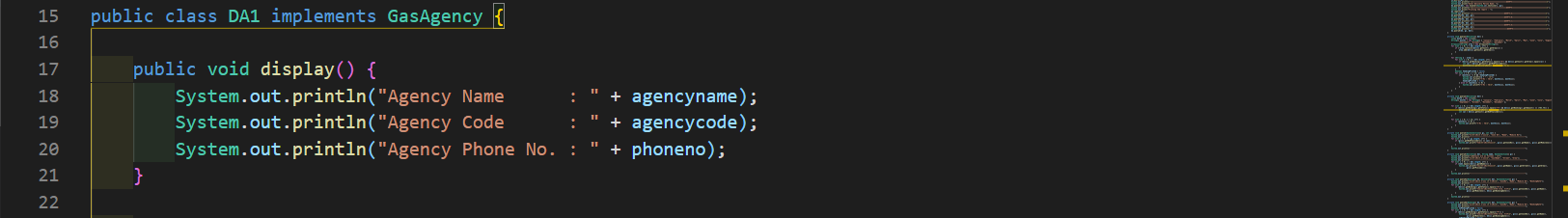
**5.PASTED CODE.**

**1.ANSWER TO THE CLASS DEFINITION PART.**

Package GAS SUPPLIER

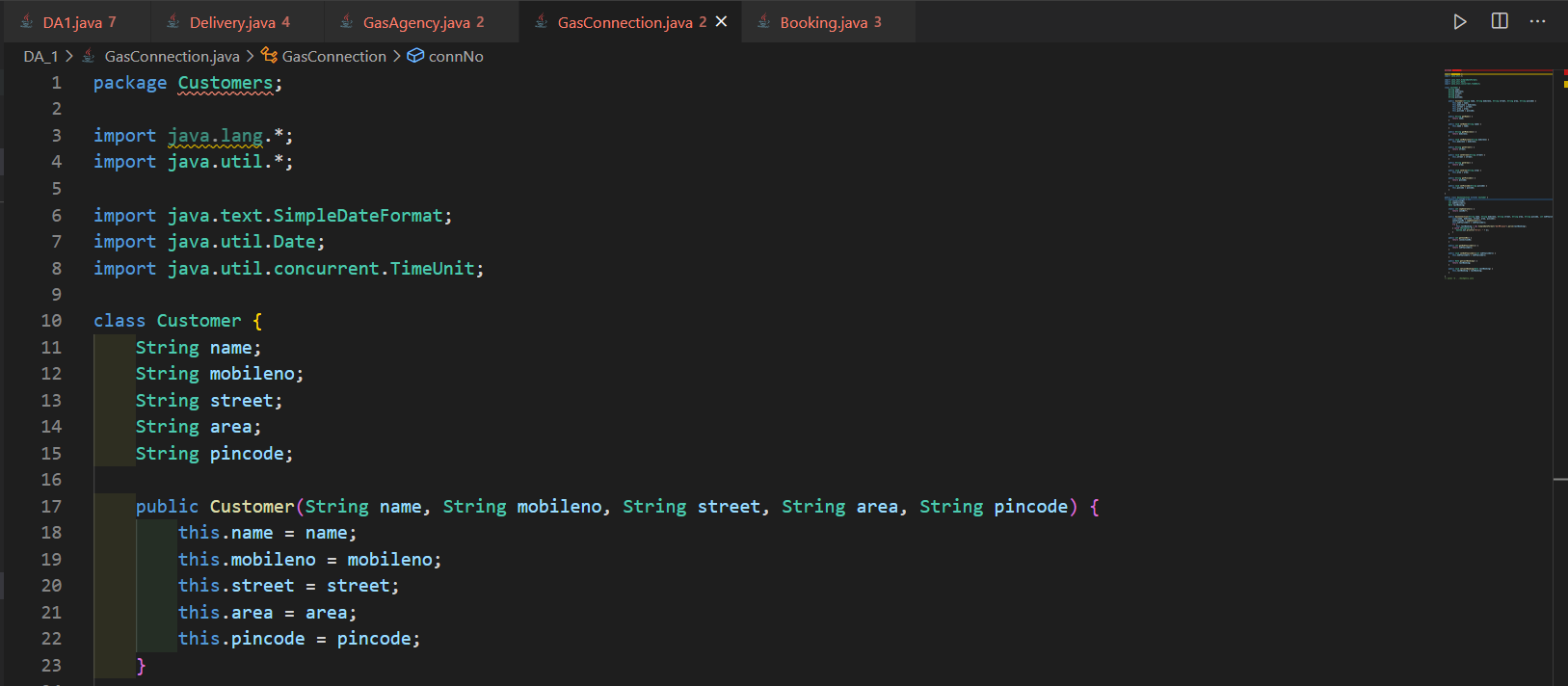
* Define an interface Gas agency which is having Agency name, agency code, phone number ( all are final variables). With the display method.





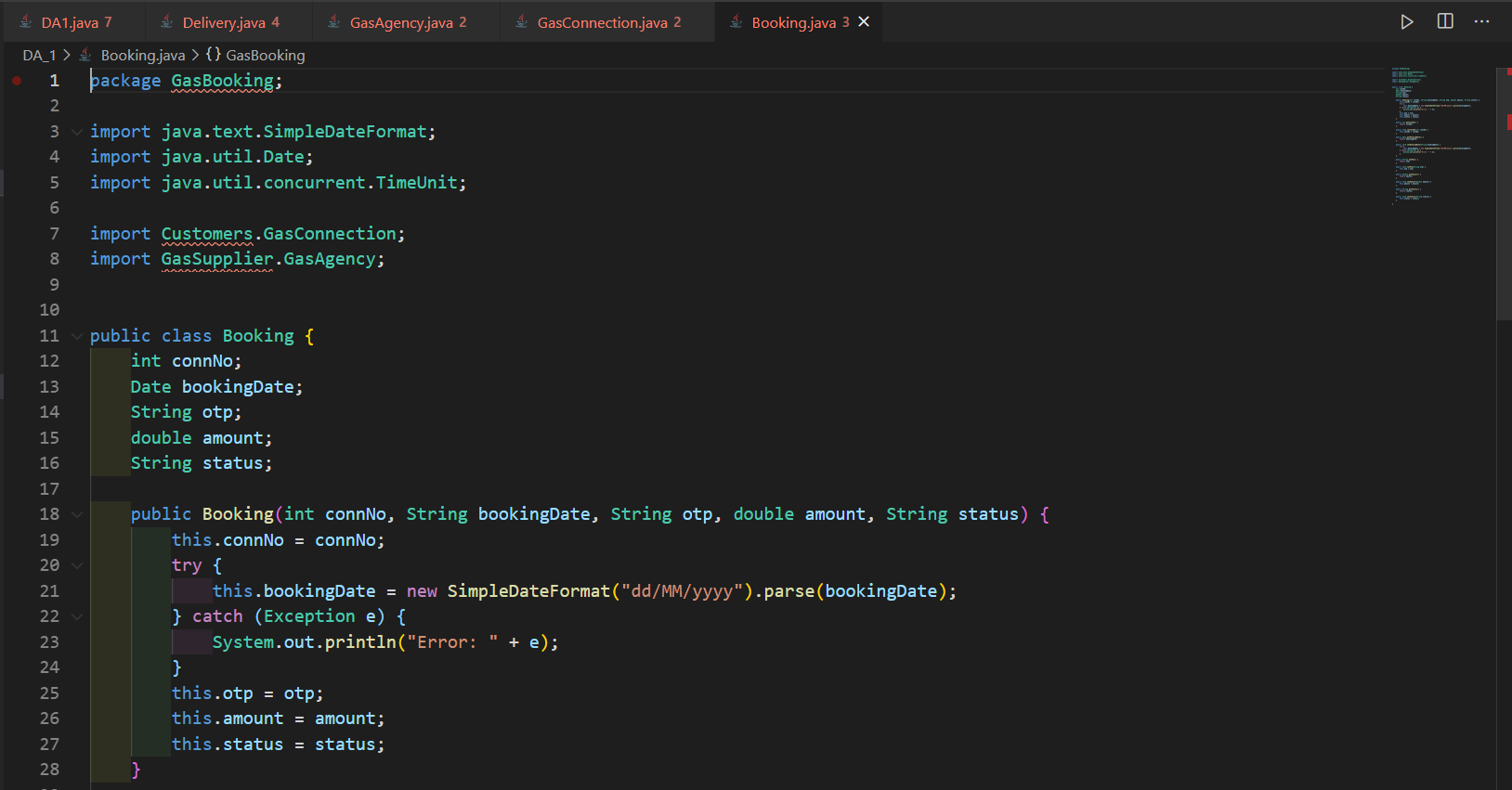
Package CUSTOMERS

* Customer class is having details like name, mobile number address ( street, area, pin code)
* Gas connection class is inherited from the customer class is having connection number (should be sequential number which is automatically generated (use static)), number of cylinders, Last booking date (current date is the last booking date while creating object).

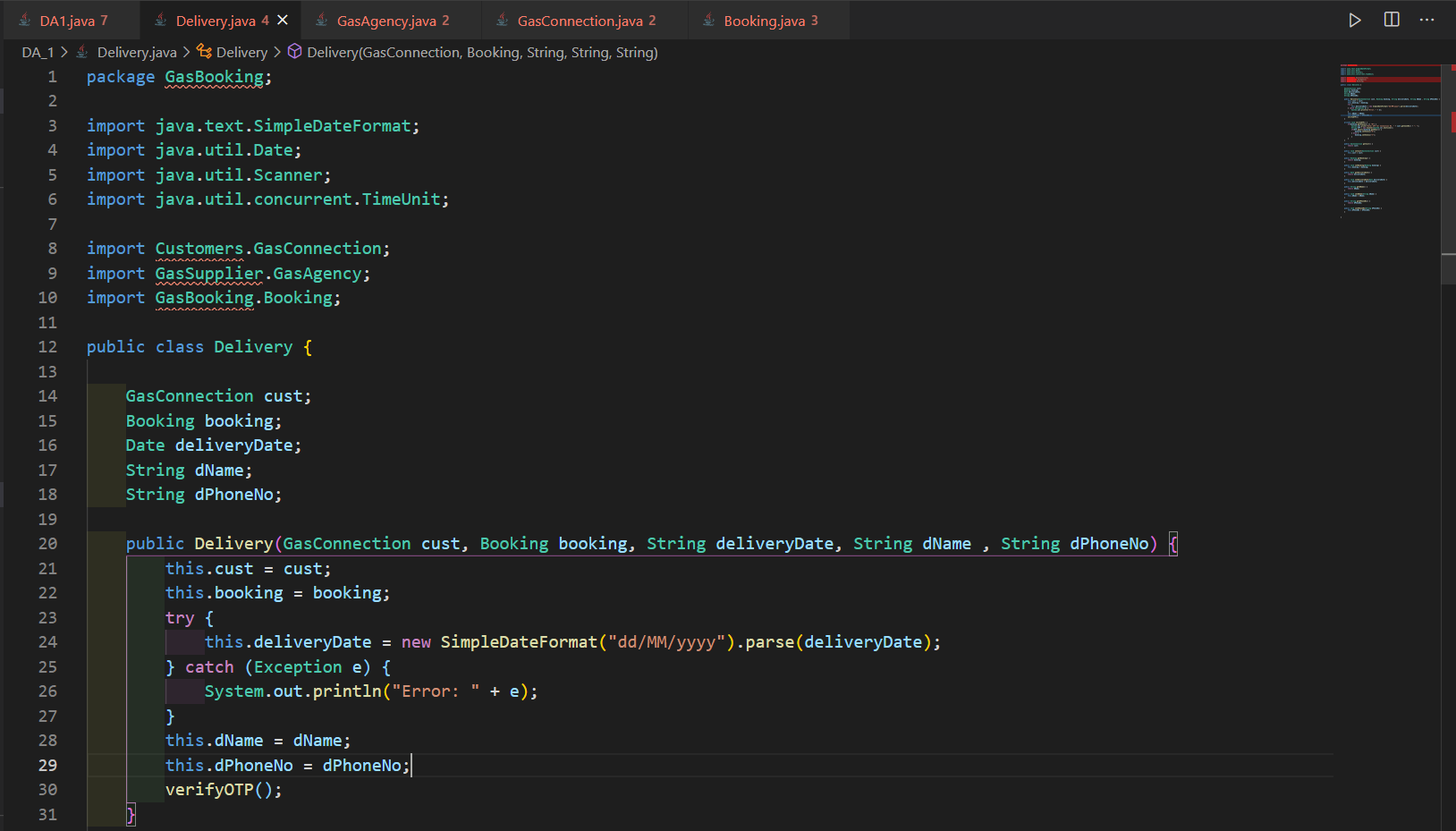


Package GASBOOKING

* Booking is the another class which records the cylinder booking details such as Gas connection number, Booking date,OTP, Amount, Status( B-Booked (during booking) , D-delivered, C-Cancelled, P-Pending) While booking validate that booking date could be 30 days from the last booking date for the single cylinder holder and 50 days for the two cylinder holder. If not booking cannot be done. If booking is done , update the last booking date in the gas connection class. After 7 days from the booking date that status is updated to (P-Pending).



* Define another class Delivery which is uses the information Gas connection number, Booking date, OTP from the booking class and have additional details like delivery date and the delivery person name.



* + While creating object for delivery person ask for the gas connection number, fetch the booking date and OTP form the booking class objects (whose status is B or P).
  + While delivery, the delivery person should ask for the OTP to the customer ( accept from the user). If the OTP given by the customer is matched with the OTP generated while booking, then only cylinder will be delivered and update the status in the booking class as (D-delivered ) else cancel the process and update the status as (C-Cancelled ). Validate that the gap between the booking date and delivery date should be between 2 to 7 days. If it exceeds 7 days, then 5% of the cylinder should be refunded to the customer. Update the amount in the booking class.

**2.SCREENSHOT OF TEST CASES.**

The test cases taken into account were :

TEST CASE 1 :

FOR (CYLHOLDER == 1){

if DIFF(LastBookingDate and CurrentBookingDate) >= 30 {

Status = = “B”; //BOOKED

} else {

Status = = “C”; //Cancelled

}

FOR (CYLHOLDER == 2){

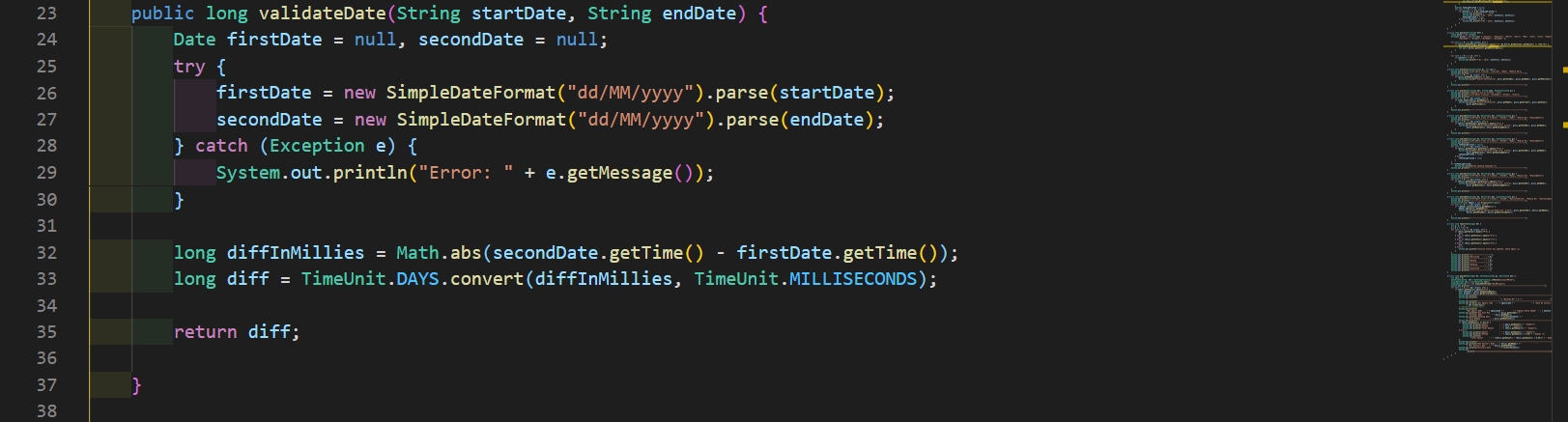
if DIFF(LastBookingDate and CurrentBookingDate) >= 70 {

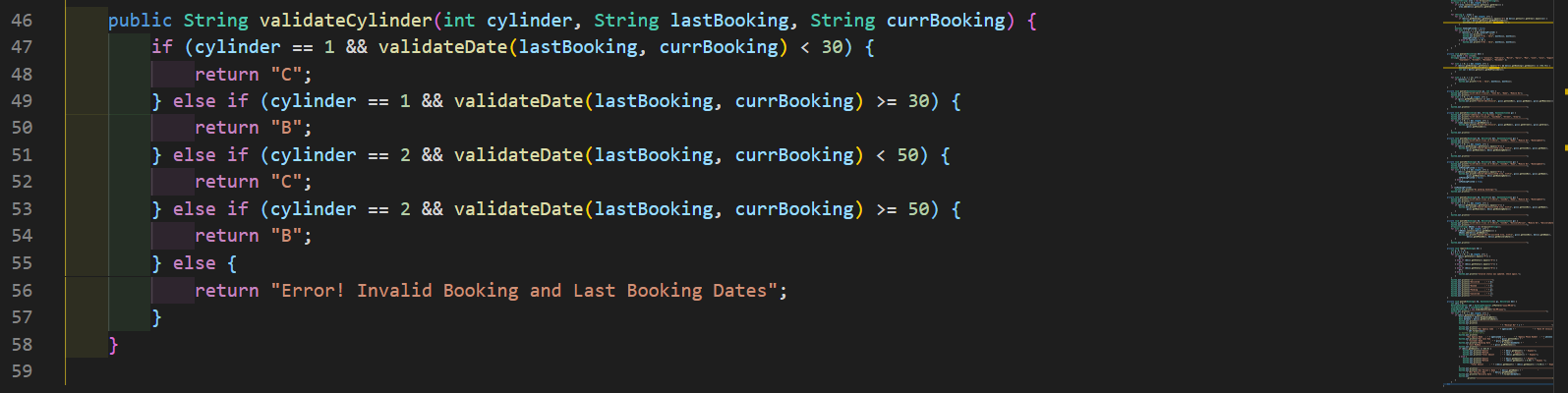
Status = = “B”; //BOOKED

} else {

Status = = “C”; //Cancelled

}



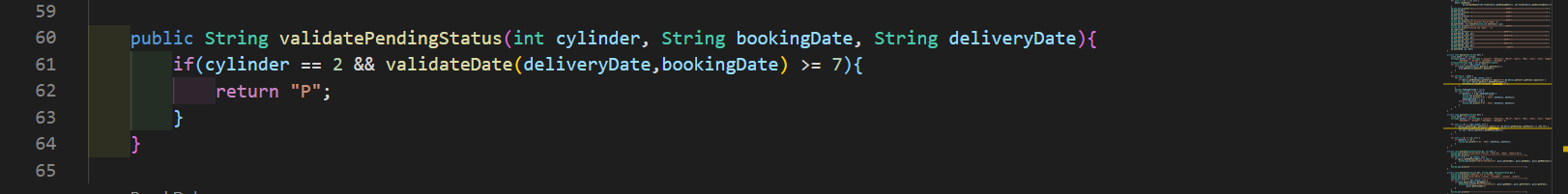


TEST CASE 2:

If DIFF(BookingDate and DeliveryDate) >= 7 {

Status = = “P”; //Pending

}



TEST CASE 3 :

Ask for OTP from the user : ------

If ( OTP == OTPGeneratedAttheTimeOfBooking){

Status = = “D” ; //Delivered

} else {

Status = = “C” ; //Cancelled

}



TEST CASE 4:

AMOUNT UPDATION

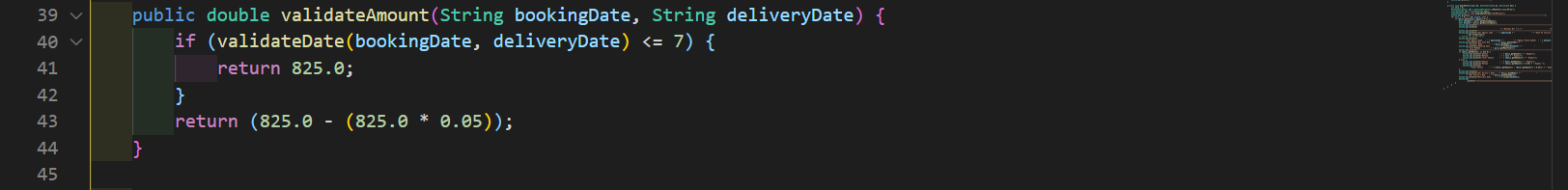
If ( DIFF( BookingDate and DeliveryDate ) > = 7) {

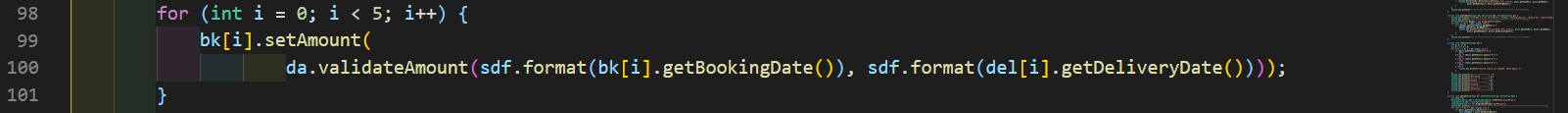
Amount = = 825 – (825\*0.05) ; // 5% Refund

} else if (DIFF( BookingDate and DeliveryDate ) < 7{

Amount = = 825 ;

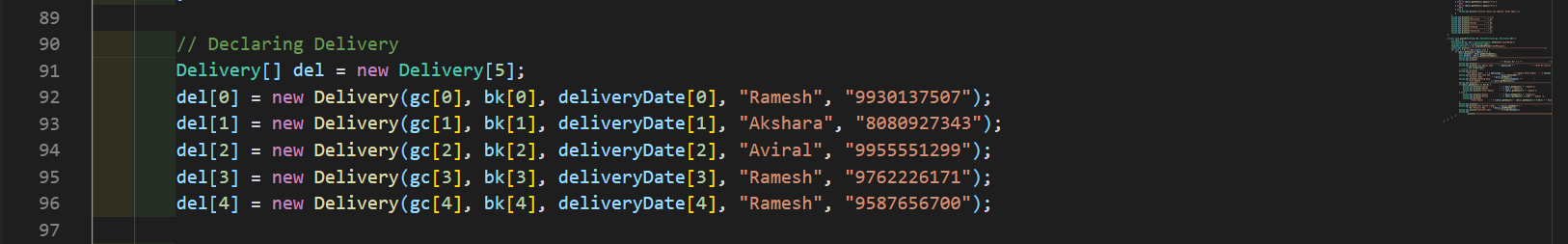
}





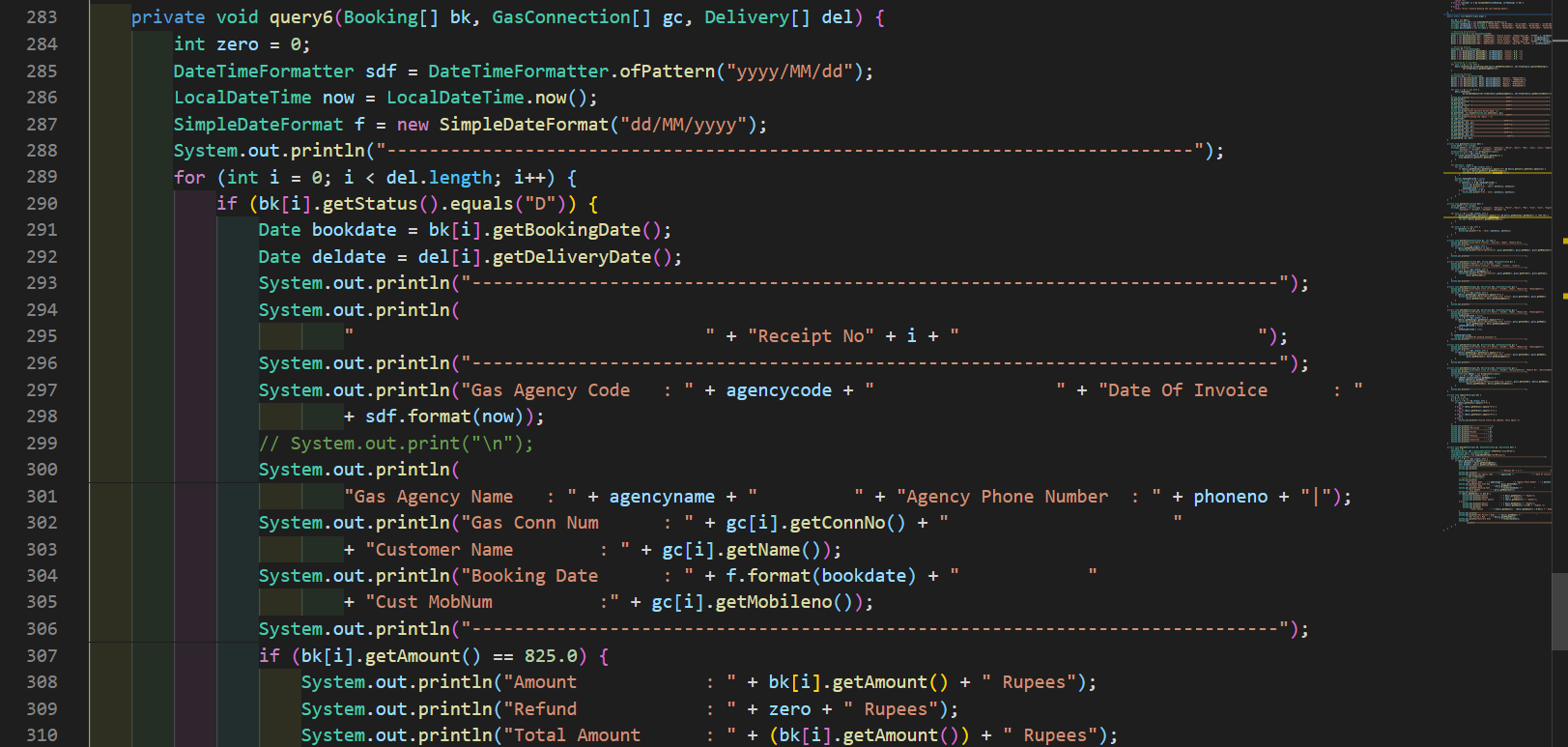
TEST CASE 5:

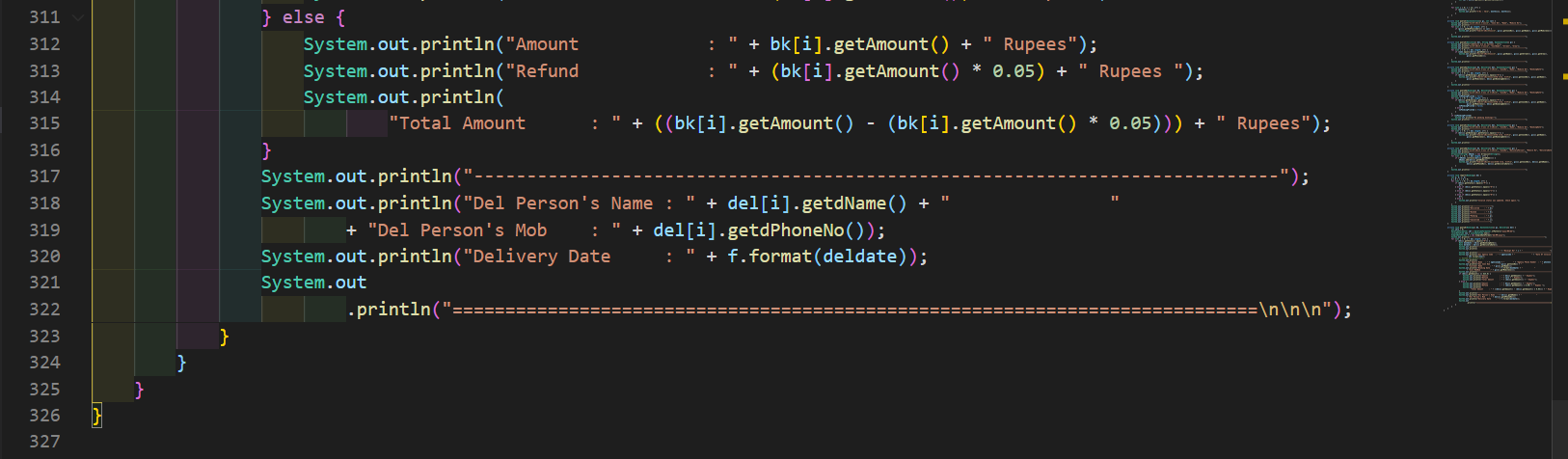
Validating Number Of Cylinders :



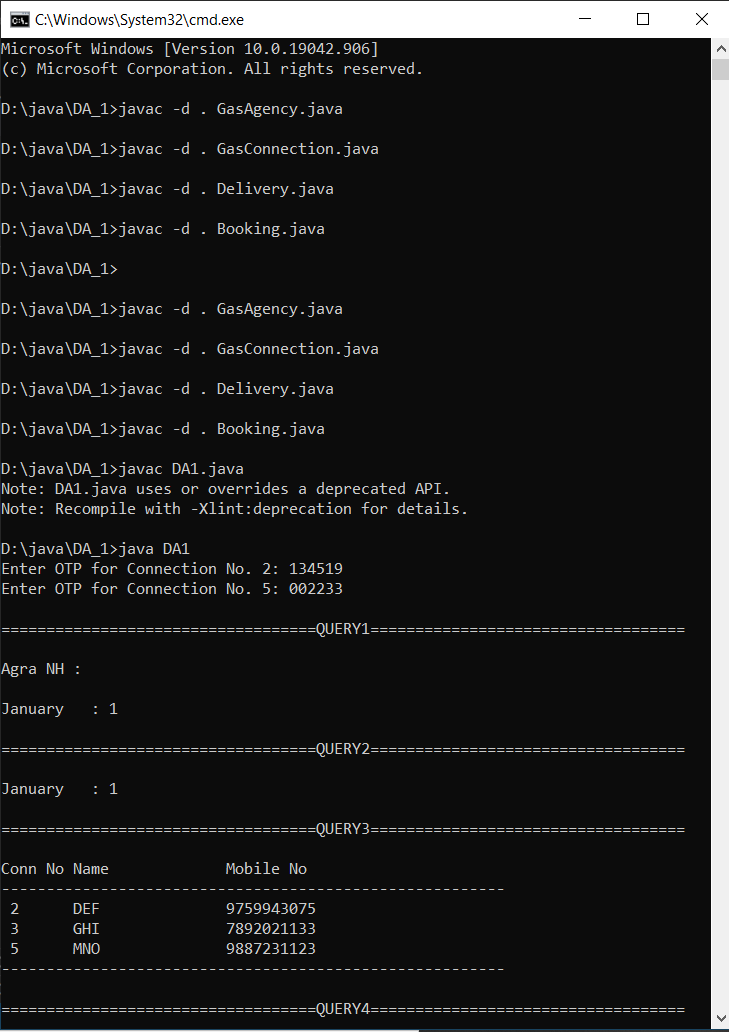
TEST CASE 6:

Printing invoice of all deliveries;

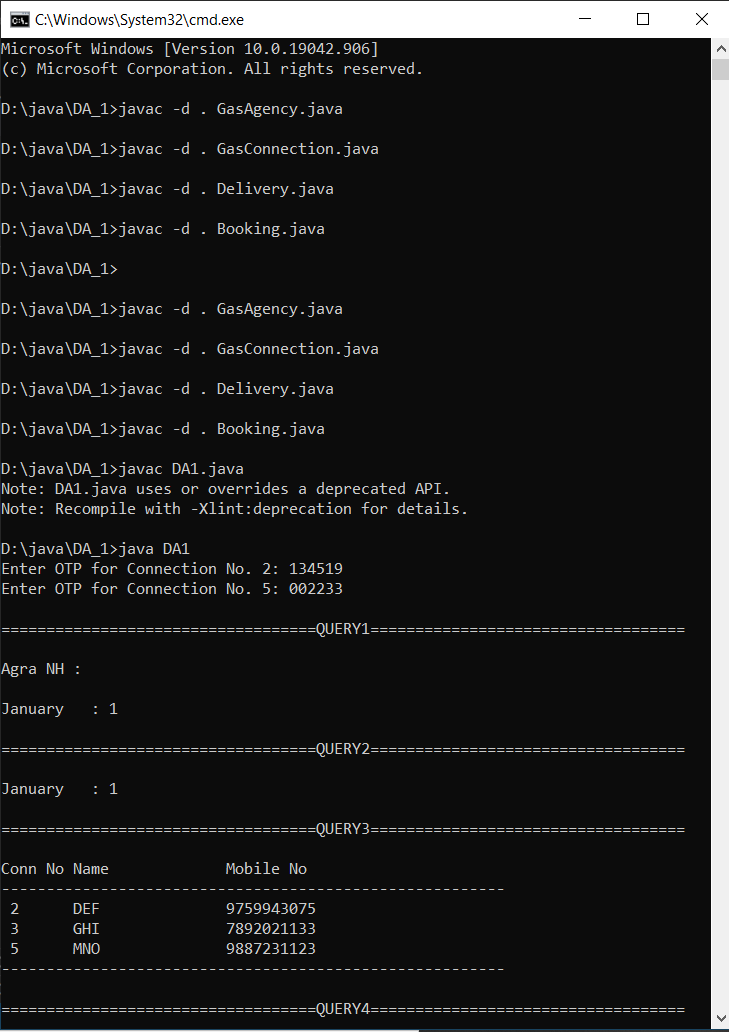




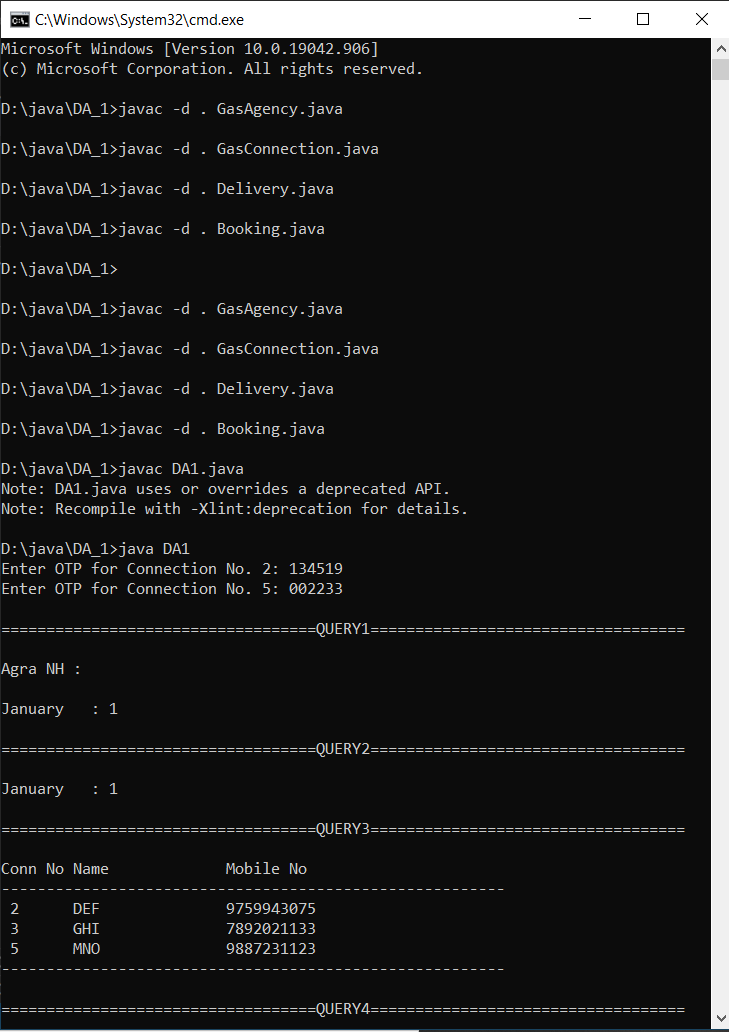
**3.SCREENSHOT OF OUTPUT.**



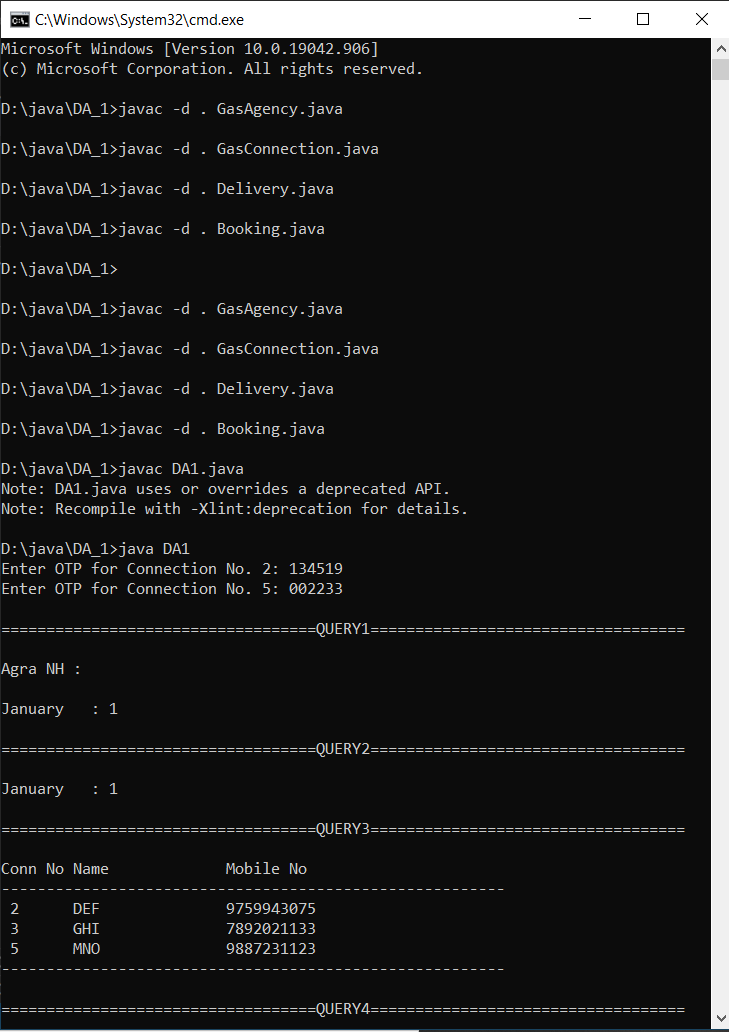
QUERY1:



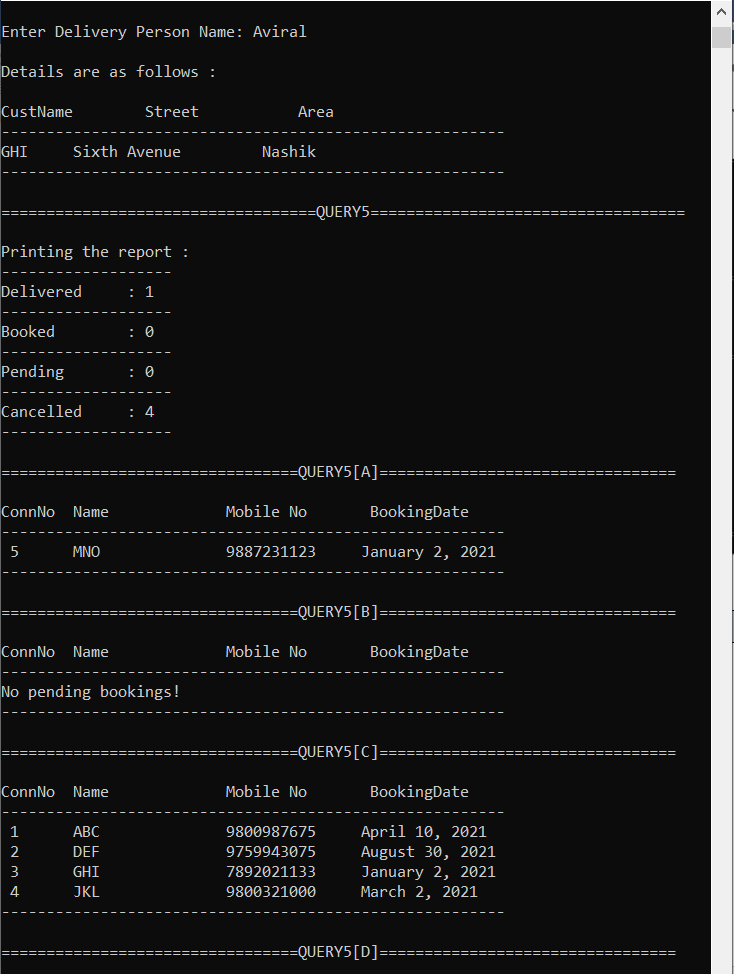
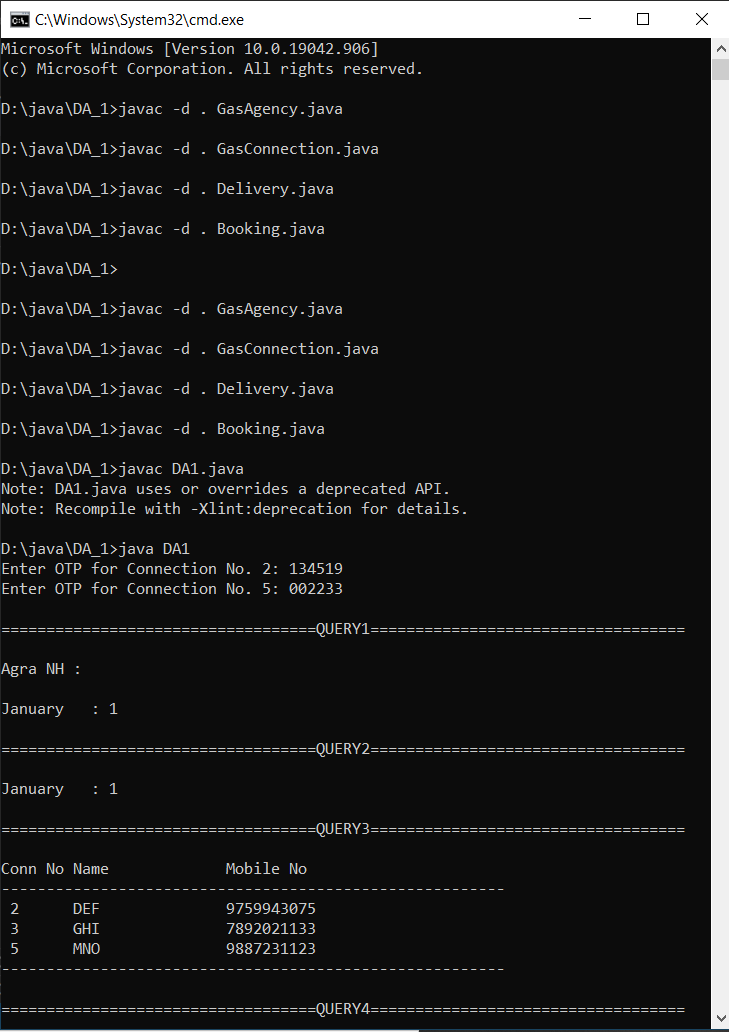
QUERY2 :



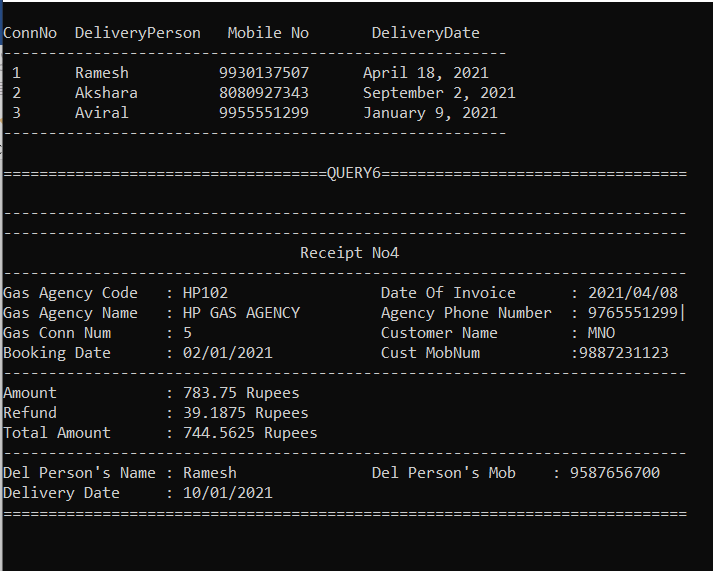
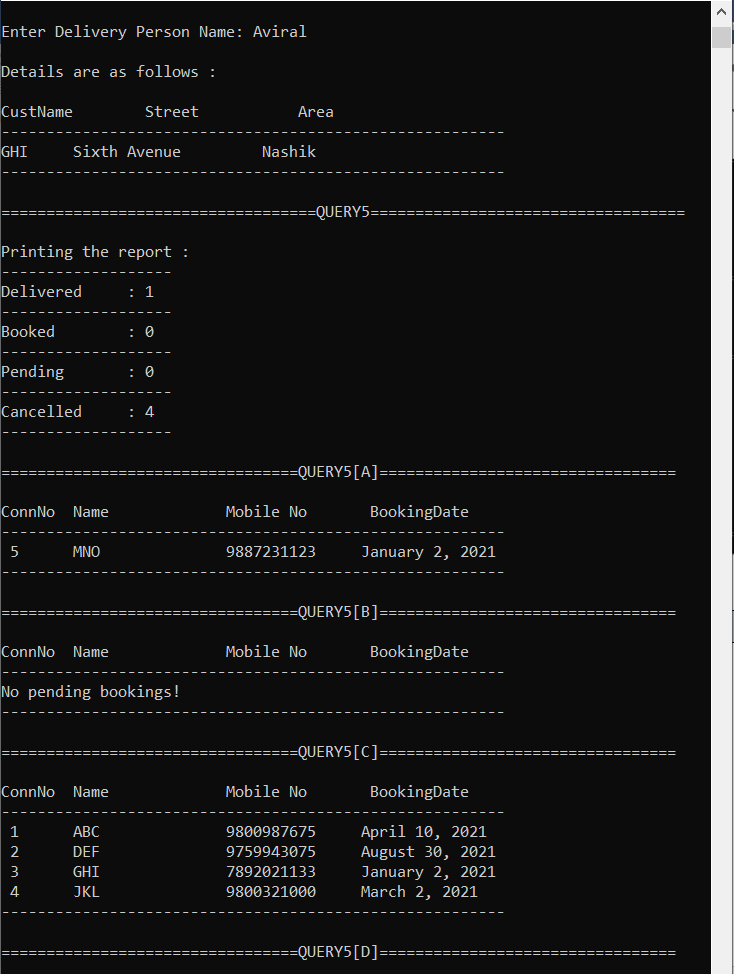
QUERY3:



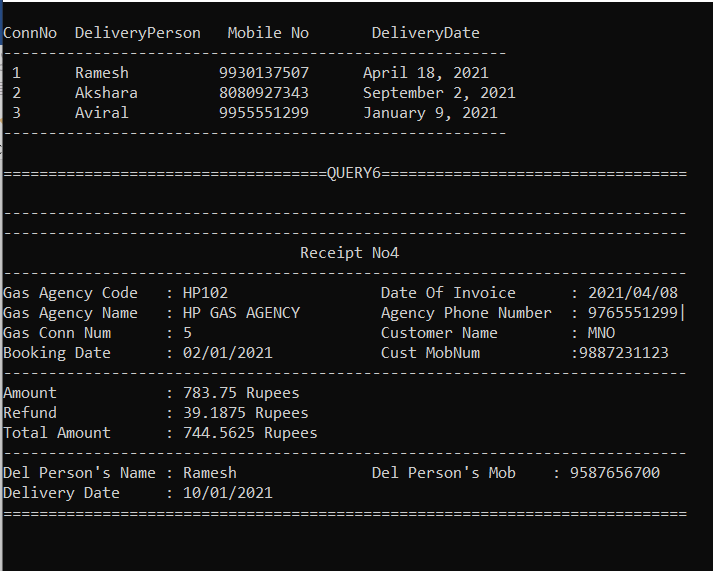
QUERY4:



QUERY5:



QUERY6: INVOICE



**4.PRE-DEFINED DATA**



**5.PASTED CODE.**

CODE:

GasAgency.java

package GasSupplier;

import java.lang.\*;

import java.util.\*;

public interface GasAgency{

    final String agencyname = "HP GAS AGENCY";

    final String agencycode = "HP102";

    final String phoneno = "9765551299";

    void display();

}

GasConnection.java

package Customers;

import java.lang.\*;

import java.util.\*;

import java.text.SimpleDateFormat;

import java.util.Date;

import java.util.concurrent.TimeUnit;

class Customer {

    String name;

    String mobileno;

    String street;

    String area;

    String pincode;

    public Customer(String name, String mobileno, String street, String area, String pincode) {

        this.name = name;

        this.mobileno = mobileno;

        this.street = street;

        this.area = area;

        this.pincode = pincode;

    }

    public String getName() {

        return name;

    }

    public void setName(String name) {

        this.name = name;

    }

    public String getMobileno() {

        return mobileno;

    }

    public void setMobileno(String mobileno) {

        this.mobileno = mobileno;

    }

    public String getStreet() {

        return street;

    }

    public void setStreet(String street) {

        this.street = street;

    }

    public String getArea() {

        return area;

    }

    public void setArea(String area) {

        this.area = area;

    }

    public String getPincode() {

        return pincode;

    }

    public void setPincode(String pincode) {

        this.pincode = pincode;

    }

}

public class GasConnection extends Customer {

    static int connNo = 1;

    int connectionNo;

    int noOfCylinders;

    Date lastBooking;

    static int seqGenerator() {

        return connNo++;

    }

    public GasConnection(String name, String mobileno, String street, String area, String pincode, int noOfCylinders, String lastBooking) {

        super(name, mobileno, street, area, pincode);

        connectionNo = seqGenerator();

        this.noOfCylinders = noOfCylinders;

        try {

            this.lastBooking = new SimpleDateFormat("dd/MM/yyyy").parse(lastBooking);

        } catch (Exception e) {

            System.out.println("Error: " + e);

        }

    }

    public int getConnNo() {

        return connectionNo;

    }

    public int getNoOfCylinders() {

        return noOfCylinders;

    }

    public void setNoOfCylinders(int noOfCylinders) {

        this.noOfCylinders = noOfCylinders;

    }

    public Date getLastBooking() {

        return lastBooking;

    }

    public void setLastBooking(Date lastBooking) {

        this.lastBooking = lastBooking;

    }

}

Booking.java

package GasBooking;

import java.text.SimpleDateFormat;

import java.util.Date;

import java.util.concurrent.TimeUnit;

import Customers.GasConnection;

import GasSupplier.GasAgency;

public class Booking {

    int connNo;

    Date bookingDate;

    String otp;

    double amount;

    String status;

    public Booking(int connNo, String bookingDate, String otp, double amount, String status) {

        this.connNo = connNo;

        try {

            this.bookingDate = new SimpleDateFormat("dd/MM/yyyy").parse(bookingDate);

        } catch (Exception e) {

            System.out.println("Error: " + e);

        }

        this.otp = otp;

        this.amount = amount;

        this.status = status;

    }

    public int getConnNo() {

        return connNo;

    }

    public void setConnNo(int connNo) {

        this.connNo = connNo;

    }

    public Date getBookingDate() {

        return bookingDate;

    }

    public void setBookingDate(String bookingDate) {

        try {

            this.bookingDate = new SimpleDateFormat("dd/MM/yyyy").parse(bookingDate);

        } catch (Exception e) {

            System.out.println("Error: " + e);

        }

    }

    public String getOtp() {

        return otp;

    }

    public void setOtp(String otp) {

        this.otp = otp;

    }

    public double getAmount() {

        return amount;

    }

    public void setAmount(double amount) {

        this.amount = amount;

    }

    public String getStatus() {

        return status;

    }

    public void setStatus(String status) {

        this.status = status;

    }

}

Delivery.java

package GasBooking;

import java.text.SimpleDateFormat;

import java.util.Date;

import java.util.Scanner;

import java.util.concurrent.TimeUnit;

import Customers.GasConnection;

import GasSupplier.GasAgency;

import GasBooking.Booking;

public class Delivery {

    GasConnection cust;

    Booking booking;

    Date deliveryDate;

    String dName;

    String dPhoneNo;

    public Delivery(GasConnection cust, Booking booking, String deliveryDate, String dName , String dPhoneNo) {

        this.cust = cust;

        this.booking = booking;

        try {

            this.deliveryDate = new SimpleDateFormat("dd/MM/yyyy").parse(deliveryDate);

        } catch (Exception e) {

            System.out.println("Error: " + e);

        }

        this.dName = dName;

        this.dPhoneNo = dPhoneNo;

        verifyOTP();

    }

    private void verifyOTP() {

        if(booking.getStatus() == "B") {

            System.out.print("Enter OTP for Connection No. " + cust.getConnNo() + ": ");

            String OTP = new Scanner(System.in).nextLine();

            if(OTP.equals(booking.getOtp())) {

                booking.setStatus("D");

            } else {

                booking.setStatus("C");

            }

        }

    }

    public GasConnection getCust() {

        return cust;

    }

    public void setCust(GasConnection cust) {

        this.cust = cust;

    }

    public Booking getBooking() {

        return booking;

    }

    public void setBooking(Booking booking) {

        this.booking = booking;

    }

    public Date getDeliveryDate() {

        return deliveryDate;

    }

    public void setDeliveryDate(Date deliveryDate) {

        this.deliveryDate = deliveryDate;

    }

    public String getdName() {

        return dName;

    }

    public void setdName(String dName) {

        this.dName = dName;

    }

    public String getdPhoneNo() {

        return dPhoneNo;

    }

    public void setdPhoneNo(String dPhoneNo) {

        this.dPhoneNo = dPhoneNo;

    }

}

DA1.java

import java.text.SimpleDateFormat;

import java.time.LocalDateTime;

import java.time.format.DateTimeFormatter;

import java.util.ArrayList;

import java.util.Date;

import java.util.Scanner;

import java.util.concurrent.TimeUnit;

import Customers.GasConnection;

import GasSupplier.GasAgency;

import GasBooking.Delivery;

import GasBooking.Booking;

public class DA1 implements GasAgency {

    public void display() {

        System.out.println("Agency Name      : " + agencyname);

        System.out.println("Agency Code      : " + agencycode);

        System.out.println("Agency Phone No. : " + phoneno);

    }

    public long validateDate(String startDate, String endDate) {

        Date firstDate = null, secondDate = null;

        try {

            firstDate = new SimpleDateFormat("dd/MM/yyyy").parse(startDate);

            secondDate = new SimpleDateFormat("dd/MM/yyyy").parse(endDate);

        } catch (Exception e) {

            System.out.println("Error: " + e.getMessage());

        }

        long diffInMillies = Math.abs(secondDate.getTime() - firstDate.getTime());

        long diff = TimeUnit.DAYS.convert(diffInMillies, TimeUnit.MILLISECONDS);

        return diff;

    }

    public double validateAmount(String bookingDate, String deliveryDate) {

        if (validateDate(bookingDate, deliveryDate) <= 7) {

            return 825.0;

        }

        return (825.0 - (825.0 \* 0.05));

    }

    public String validateCylinder(int cylinder, String lastBooking, String currBooking) {

        if (cylinder == 1 && validateDate(lastBooking, currBooking) < 30) {

            return "C";

        } else if (cylinder == 1 && validateDate(lastBooking, currBooking) >= 30) {

            return "B";

        } else if (cylinder == 2 && validateDate(lastBooking, currBooking) < 50) {

            return "C";

        } else if (cylinder == 2 && validateDate(lastBooking, currBooking) >= 50) {

            return "B";

        } else {

            return "Error! Invalid Booking and Last Booking Dates";

        }

    }

    public static void main(String[] args) {

        DA1 da = new DA1();

        SimpleDateFormat sdf = new SimpleDateFormat("dd/MM/yyyy");

        String[] lastBooking = new String[] { "03/04/2021", "01/07/2021", "31/12/2020", "12/02/2021", "12/08/2020" };

        String[] currBooking = new String[] { "10/04/2021", "30/08/2021", "02/01/2021", "02/03/2021", "02/01/2021" };

        String[] deliveryDate = new String[] { "18/04/2021", "02/09/2021", "09/01/2021", "11/03/2021", "10/01/2021" };

        // Declaring GasConnection

        GasConnection[] gc = new GasConnection[5];

        gc[0] = new GasConnection("ABC", "9800987675", "Fourth Avenue", "Mumbai-Agra NH", "422009", 2, lastBooking[0]);

        gc[1] = new GasConnection("DEF", "9759943075", "Fifth Avenue", "Nashik", "423009", 1, lastBooking[1]);

        gc[2] = new GasConnection("GHI", "7892021133", "Sixth Avenue", "Nashik", "423009", 1, lastBooking[2]);

        gc[3] = new GasConnection("JKL", "9800321000", "Fourth Avenue", "Mumbai-Agra NH", "422009", 2, lastBooking[3]);

        gc[4] = new GasConnection("MNO", "9887231123", "Fifth Avenue", "Agra NH", "422349", 1, lastBooking[4]);

        // Declaring Booking

        Booking[] bk = new Booking[5];

        bk[0] = new Booking(gc[0].getConnNo(), currBooking[0], "111111", 0.0, "");

        bk[1] = new Booking(gc[1].getConnNo(), currBooking[1], "111111", 0.0, "");

        bk[2] = new Booking(gc[2].getConnNo(), currBooking[2], "111111", 0.0, "");

        bk[3] = new Booking(gc[3].getConnNo(), currBooking[3], "111111", 0.0, "");

        bk[4] = new Booking(gc[4].getConnNo(), currBooking[4], "111111", 0.0, "");

        // Verifing No of Cylinders

        for (int i = 0; i < 5; i++) {

            bk[i].setStatus(da.validateCylinder(gc[i].getNoOfCylinders(), sdf.format(gc[i].getLastBooking()),

                    sdf.format(bk[i].getBookingDate())));

        }

        // Declaring Delivery

        Delivery[] del = new Delivery[5];

        del[0] = new Delivery(gc[0], bk[0], deliveryDate[0], "Ramesh", "9930137507");

        del[1] = new Delivery(gc[1], bk[1], deliveryDate[1], "Akshara", "8080927343");

        del[2] = new Delivery(gc[2], bk[2], deliveryDate[2], "Aviral", "9955551299");

        del[3] = new Delivery(gc[3], bk[3], deliveryDate[3], "Ramesh", "9762226171");

        del[4] = new Delivery(gc[4], bk[4], deliveryDate[4], "Ramesh", "9587656700");

        for (int i = 0; i < 5; i++) {

            bk[i].setAmount(

                    da.validateAmount(sdf.format(bk[i].getBookingDate()), sdf.format(del[i].getDeliveryDate())));

        }

        System.out.println("\n===================================QUERY1===================================\n");

        da.query1(del);

        System.out.println("\n===================================QUERY2===================================\n");

        da.query2(del);

        System.out.println("\n===================================QUERY3===================================\n");

        da.query3(gc, 1);

        System.out.println("\n===================================QUERY4===================================\n");

        System.out.print("Enter Delivery Person Name: ");

        da.query4(del, new Scanner(System.in).nextLine(), gc);

        System.out.println("\n===================================QUERY5===================================\n");

        System.out.print("Printing the report : ");

        da.report(bk);

        System.out.println("\n=================================QUERY5[A]=================================\n");

        da.query5a(bk, del, gc);

        System.out.println("\n=================================QUERY5[B]=================================\n");

        da.query5b(bk, del, gc);

        System.out.println("\n=================================QUERY5[C]=================================\n");

        da.query5c(bk, del, gc);

        System.out.println("\n=================================QUERY5[D]=================================\n");

        da.query5d(bk, del, gc);

        System.out.println("\n====================================QUERY6==================================\n");

        da.query6(bk, gc, del);

    }

    private void query1(Delivery[] del) {

        int[] month = new int[12];

        String[] months = new String[] { "January", "February", "March", "April", "May", "June", "July", "August",

                "September", "October", "November", "December" };

        ArrayList<String> area = new ArrayList<String>();

        for (int i = 0; i < del.length; i++) {

            if (!area.contains(del[i].getCust().getArea())) {

                area.add(del[i].getCust().getArea());

            }

        }

        for (String a : area) {

            for (int i = 0; i < del.length; i++) {

                if (del[i].getBooking().getStatus().equals("D") && del[i].getCust().getArea().equals(a)) {

                    int cyl = del[i].getCust().getNoOfCylinders();

                    month[del[i].getDeliveryDate().getMonth()] += 1;

                }

            }

            boolean headingPrinted = false;

            for (int i = 0; i < 12; i++) {

                if (month[i] != 0 && !headingPrinted) {

                    System.out.println(a + " : \n");

                    System.out.printf("%-9s : %d\n", months[i], month[i]);

                    headingPrinted = true;

                } else if (month[i] != 0) {

                    System.out.printf("%-9s : %d\n", months[i], month[i]);

                }

            }

        }

    }

    private void query2(Delivery[] del) {

        int[] month = new int[12];

        String[] months = new String[] { "January", "February", "March", "April", "May", "June", "July", "August",

                "September", "October", "November", "December" };

        for (int i = 0; i < del.length; i++) {

            if (del[i].getBooking().getStatus().equals("D") && del[i].getBooking().getAmount() == (783.75)) {

                month[del[i].getDeliveryDate().getMonth()] += 1;

                int cyl = del[i].getCust().getNoOfCylinders();

            }

        }

        for (int i = 0; i < 12; i++) {

            if (month[i] != 0) {

                System.out.printf("%-9s : %d\n", months[i], month[i]);

            }

        }

    }

    private void query3(GasConnection[] gc, int cyl) {

        System.out.printf("%s\t%-10s\t %-11s\n", "Conn No", "Name", "Mobile No");

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

        for (int i = 0; i < gc.length; i++) {

            if (gc[i].getNoOfCylinders() == cyl) {

                System.out.printf("%2d\t%-10s\t%11s\n", gc[i].getConnNo(), gc[i].getName(), gc[i].getMobileno());

            }

        }

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

    }

    private void query4(Delivery[] del, String name, GasConnection[] gc) {

        System.out.println("\nDetails are as follows : \n");

        System.out.printf("%s\t%-10s\t %-11s\n", "CustName", "Street", "Area");

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

        for (int i = 0; i < del.length; i++) {

            if (name.equals(del[i].getdName())) {

                System.out.printf("%2s\t%-10s\t%11s\n", gc[i].getName(), gc[i].getStreet(), gc[i].getArea(),

                        gc[i].getPincode());

            }

        }

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

    }

    private void query5a(Booking[] bk, Delivery[] del, GasConnection[] gc) {

        System.out.printf("%s\t%-10s\t %-11s \t %-10s\n", "ConnNo", "Name", "Mobile No", "BookingDate");

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

        for (int i = 0; i < del.length; i++) {

            if (del[i].getBooking().getStatus().equals("D")) {

                System.out.printf("%2d\t%-10s\t%11s\t%tB %<te, %<tY\n", gc[i].getConnNo(), gc[i].getName(),

                        gc[i].getMobileno(), bk[i].getBookingDate());

            }

        }

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

    }

    private void query5b(Booking[] bk, Delivery[] del, GasConnection[] gc) {

        System.out.printf("%s\t%-10s\t %-11s \t %-10s\n", "ConnNo", "Name", "Mobile No", "BookingDate");

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

        boolean noPendingPrinted = false;

        for (int i = 0; i < del.length; i++) {

            if (del[i].getBooking().getStatus().equals("P")) {

                System.out.printf("%2d\t%-10s\t%11s\t%tB %<te, %<tY\n", gc[i].getConnNo(), gc[i].getName(),

                        gc[i].getMobileno(), bk[i].getBookingDate());

                noPendingPrinted = false;

            } else {

                noPendingPrinted = true;

            }

        }

        if (noPendingPrinted)

            System.out.println("No pending bookings!");

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

    }

    private void query5c(Booking[] bk, Delivery[] del, GasConnection[] gc) {

        System.out.printf("%s\t%-10s\t %-11s \t %-10s\n", "ConnNo", "Name", "Mobile No", "BookingDate");

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

        for (int i = 0; i < del.length; i++) {

            if (del[i].getBooking().getStatus().equals("C")) {

                System.out.printf("%2d\t%-10s\t%11s\t%tB %<te, %<tY\n", gc[i].getConnNo(), gc[i].getName(),

                        gc[i].getMobileno(), bk[i].getBookingDate());

            }

        }

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

    }

    private void query5d(Booking[] bk, Delivery[] del, GasConnection[] gc) {

        System.out.printf("%s\t%-10s\t %-11s \t %-10s\n", "ConnNo", "DeliveryPerson", "Mobile No", "DeliveryDate");

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

        ArrayList<String> dNames = new ArrayList<String>();

        for (int i = 0; i < del.length; i++) {

            if (!dNames.contains(del[i].getdName())) {

                dNames.add(del[i].getdName());

                System.out.printf("%2d\t%-10s\t%-11s\t%tB %<te, %<tY\n", gc[i].getConnNo(), del[i].getdName(),

                        del[i].getdPhoneNo(), del[i].getDeliveryDate());

            }

        }

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

    }

    private void report(Booking[] bk) {

        int p, d, c, b;

        p = d = c = b = 0;

        for (int i = 0; i < bk.length; i++) {

            if (bk[i].getStatus().equals("P")) {

                p++;

            } else if (bk[i].getStatus().equals("D")) {

                d++;

            } else if (bk[i].getStatus().equals("C")) {

                c++;

            } else if (bk[i].getStatus().equals("B")) {

                b++;

            } else {

                System.out.println("Invalid status was updated. Check again.");

            }

        }

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

        System.out.println("Delivered     : " + d);

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

        System.out.println("Booked        : " + b);

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

        System.out.println("Pending       : " + p);

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

        System.out.println("Cancelled     : " + c);

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

    }

    private void query6(Booking[] bk, GasConnection[] gc, Delivery[] del) {

        int zero = 0;

        DateTimeFormatter sdf = DateTimeFormatter.ofPattern("yyyy/MM/dd");

        LocalDateTime now = LocalDateTime.now();

        SimpleDateFormat f = new SimpleDateFormat("dd/MM/yyyy");

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

        for (int i = 0; i < del.length; i++) {

            if (bk[i].getStatus().equals("D")) {

                Date bookdate = bk[i].getBookingDate();

                Date deldate = del[i].getDeliveryDate();

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

                System.out.println(

                        "                                 " + "Receipt No" + i + "                            ");

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

                System.out.println("Gas Agency Code   : " + agencycode + "                 " + "Date Of Invoice      : "

                        + sdf.format(now));

                // System.out.print("\n");

                System.out.println(

                        "Gas Agency Name   : " + agencyname + "         " + "Agency Phone Number  : " + phoneno + "|");

                System.out.println("Gas Conn Num      : " + gc[i].getConnNo() + "                     "

                        + "Customer Name        : " + gc[i].getName());

                System.out.println("Booking Date      : " + f.format(bookdate) + "            "

                        + "Cust MobNum          :" + gc[i].getMobileno());

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

                if (bk[i].getAmount() == 825.0) {

                    System.out.println("Amount            : " + bk[i].getAmount() + " Rupees");

                    System.out.println("Refund            : " + zero + " Rupees");

                    System.out.println("Total Amount      : " + (bk[i].getAmount()) + " Rupees");

                } else {

                    System.out.println("Amount            : " + bk[i].getAmount() + " Rupees");

                    System.out.println("Refund            : " + (bk[i].getAmount() \* 0.05) + " Rupees ");

                    System.out.println(

                            "Total Amount      : " + ((bk[i].getAmount() - (bk[i].getAmount() \* 0.05))) + " Rupees");

                }

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

                System.out.println("Del Person's Name : " + del[i].getdName() + "               "

                        + "Del Person's Mob    : " + del[i].getdPhoneNo());

                System.out.println("Delivery Date     : " + f.format(deldate));

                System.out

                        .println("============================================================================\n\n\n");

            }

        }

    }

}