

Informatics Institute of Technology
School of Computing
Software Development II Coursework Report

Module : 4COSC010C.2: Software Development II (2023)

Date of submission : 25.03.2024

Student ID : 20231979 / w2051998

Student First Name : Akaash

Student Surname : Sashiraj

Tutorial group (day, time, and tutor/s):

Group : 7

Day : Wednesday

Time : 1.30pm – 3.30pm

Tutors: Dimithri Premachandra

Dinusha Ruwan Kumara

"I confirm that I understand what plagiarism / collusion / contract cheating is and have read and understood the section on Assessment Offences in the Essential Information for Students. The work that I have submitted is entirely my own. Any work from other authors is duly referenced and acknowledged."

Name : Sashiraj Akaash

Student ID : IIT No- 20231979 | UOW No – w2051998

Self-assessment form and test plan

1) Self-assessment form

Task	Self-assessment (select one)	Comments
1	<input checked="" type="checkbox"/> Fully implemented <input type="checkbox"/> Partially implemented <input type="checkbox"/> Not attempted	Here a 2D array was used for the seating plan. At the start of this program all seats are should be available for that I used Enhanced for-loop to assigned zero for seats.
2	<input checked="" type="checkbox"/> Fully implemented <input type="checkbox"/> Partially implemented <input type="checkbox"/> Not attempted	This menu option is created as a separate method called menu(). And user in input 0, When program ask to select the option the program will terminated.
Insert here a screenshot of your welcome message and menu: <div style="background-color: #2e3436; color: #eeeeec; padding: 10px; margin: 10px 0;"> <pre> ***** * Menu Option * ***** 1) Buy a seat 2) Cancel a seat 3) Find first seat available 4) Show seating plan 5) Print tickets information and total sales 6) Search tickets 0) Quit ***** Select your option (0-6): </pre> </div>		
3	<input checked="" type="checkbox"/> Fully implemented <input type="checkbox"/> Partially implemented <input type="checkbox"/> Not attempted	The buy_seat method asks the user to enter a seat number and a row letter (A–D), validates the input, determines whether the seat is available, marks the seat as sold. Error messages are displayed in response to invalid input and unavailable seats.

4	<input checked="" type="checkbox"/> Fully implemented <input type="checkbox"/> Partially implemented <input type="checkbox"/> Not attempted	If a reserved seat is available, the cancel_seat function requests the row letter and seat number from the user, cancels it, and verifies the cancellation. The user is notified if the assigned seat is not reserved.
5	<input checked="" type="checkbox"/> Fully implemented <input type="checkbox"/> Partially implemented <input type="checkbox"/> Not attempted	The find_first_available function loops through the seat array in search of the first open seat on the plane. The first available seat's row letter and seat number are printed. It prints a notice stating that all tickets are reserved if there are no seats available .here used nested for-loop for check the availability of seats in 2D array
6	<input checked="" type="checkbox"/> Fully implemented <input type="checkbox"/> Partially implemented <input type="checkbox"/> Not attempted	The plane's seating arrangement is shown using the show_seating_plan method. using an enhanced loop., it iterates through each row of seats, printing the row label ('A', 'B', 'C', etc.) and the seat status ('0' for available, 'X' for booked).

Insert here a screenshot of the seating plan:

```
Select your option (0-6):
4
A 0 0 0 0 0 0 0 0 0 0 0 0 0 0
B 0 0 0 0 0 0 0 0 0 0 0 0 0
C 0 0 0 0 0 0 0 0 0 0 0 0
D 0 0 0 0 0 0 0 0 0 0 0 0 0
```

7	<input checked="" type="checkbox"/> Fully implemented <input type="checkbox"/> Partially implemented <input type="checkbox"/> Not attempted	Print method print the information input by the passenger by taking them from getters.
8	<input checked="" type="checkbox"/> Fully implemented <input type="checkbox"/> Partially implemented <input type="checkbox"/> Not attempted	Information about tickets, including row letter, seat number, price, and passenger details, are managed by the Tickets class. It provides ways to obtain and modify these properties. The TicketDetails method prints the ticket information with passenger detail
9	<input checked="" type="checkbox"/> Fully implemented <input type="checkbox"/> Partially implemented <input type="checkbox"/> Not attempted	Ticket and user's input will added to the array.
10	<input checked="" type="checkbox"/> Fully implemented <input type="checkbox"/> Partially implemented <input type="checkbox"/> Not attempted	The method outputs the information about each booked seat, iterates through them, and determines the total cost of all booked tickets. To access ticket objects stored in a 2D array and display ticket details, it runs the TicketDetails function using nested loops.
11	<input checked="" type="checkbox"/> Fully implemented <input type="checkbox"/> Partially implemented <input type="checkbox"/> Not attempted	By entering the row letter (A–D) and seat number, the user can search for a booked ticket using this method. The user is notified that the seat is available for booking if there is no ticket at the indicated location; if there is, its details are provided.
12	<input checked="" type="checkbox"/> Fully implemented <input type="checkbox"/> Partially implemented <input type="checkbox"/> Not attempted	This method saves the passenger information and seat information that is entered by user. This create

		a external file to save the details that entered by user.
--	--	---

2) Test Plan

Complete the test plan describing which testing you have performed on your program.
Add as many rows as you need.

Part A Testing

Test case / scenario	Input	Expected Output	Output	Pass/Fail
When invalid option selected	Option = any alphabet letter or something not in 0-6	Give a message to user to enter the number between 0-6.	Prints "Incorrect input. Please enter a number (0-6):"	<input checked="" type="checkbox"/> Pass <input type="checkbox"/> Fail
When user input valid row and seat	Option= 1 Row=A Seat=3 Name=Sachin Surname=Tendulkar Email=sachin@12.com	A3 seat is booked and information saved as ticket	Prints "Your Ticket is booked. Thank you..." And also save ticket in booking information file as text file named "A3"	<input checked="" type="checkbox"/> Pass <input type="checkbox"/> Fail
When user input invalid row	Option= 1 Row=S Seat=3	Print a message to user that the input is invalid	Prints" Your selection is invalid. Please try again"	<input checked="" type="checkbox"/> Pass <input type="checkbox"/> Fail
When user input invalid seatno	Option= 1 Row=A Seat=30	Print a message to user that the input is invalid	Prints" Your selection is invalid. Please try again"	<input checked="" type="checkbox"/> Pass <input type="checkbox"/> Fail
When user not enter the name	Option= 1 Row=A Seat=3 Name=	Print a message that name cannot be empty	Prints" Passenger's Name cannot be empty. Enter passenger's name:"	<input checked="" type="checkbox"/> Pass <input type="checkbox"/> Fail

When user not enter the surname	Option= 1 Row=A Seat=3 Name=Sachin Surname=	Print a message that Surname cannot be empty	Prints" Passenger's Surname cannot be empty. Enter passenger's surname:"	<input checked="" type="checkbox"/> Pass <input type="checkbox"/> Fail
When user not enter the email	Option= 1 Row=A Seat=3 Name=Sachin Surname=Tendulkar Email=	Print a message that email cannot be empty	Prints" Passenger's Email cannot be empty. Enter passenger's Email: "	<input checked="" type="checkbox"/> Pass <input type="checkbox"/> Fail
When user enter the already booked row and seat number	Option= 1 Row=A Seat=3	Print a message that this seat is already booked.	Prints" Sorry this ticket is already booked...Try another seat to book"	<input checked="" type="checkbox"/> Pass <input type="checkbox"/> Fail
When user enter the valid row and seat to cancel the booking	Option= 2 Row=A Seat=3	Print a message that that seat booking is canceled.	Prints "Your booking is cancelled"	<input checked="" type="checkbox"/> Pass <input type="checkbox"/> Fail
When user enter the invalid row and seat to cancel the booking	Option= 2 Row=A Seat=5	Print a message that that seat is not booked yet.	Prints "This seat is not booked yet"	<input checked="" type="checkbox"/> Pass <input type="checkbox"/> Fail
When user input valid row and seat to see	Option= 3	Print the seat that available first.	Prints" First available seats: row= A Seat Number= 1"	<input checked="" type="checkbox"/> Pass <input type="checkbox"/> Fail

first available seat				
When all seat are booked	Option=3	Print a message that all seats are booked	Prints "All Tickets are booked...."	<input checked="" type="checkbox"/> Pass <input type="checkbox"/> Fail
When user want to buy another seat	Option= 1 Row=A Seat=5 Name=Virat Surname=kohli Email=virat@12.com	A15seat is booked and information saved as ticket	Prints "Your Ticket is booked. Thank you..." And also save ticket in booking information file as text file named "A5"	<input checked="" type="checkbox"/> Pass <input type="checkbox"/> Fail
When user needs to see seating plan	Option=4	Print seating plan (O O O O X O)	Print seating plan (O O O O X O) "A5" ticket is booked.	<input checked="" type="checkbox"/> Pass <input type="checkbox"/> Fail

Part B testing

Test case / scenario	Input	Expected Output	Output	Pass/Fail
When user want to buy another seat	Option= 1 Row=A Seat=5 Name=Virat Surname=kohli Email=virat@12.com	A15seat is booked and information saved as ticket	Prints "Your Ticket is booked. Thank you..." And also save ticket in booking information file as text file named "A5"	<input checked="" type="checkbox"/> Pass <input type="checkbox"/> Fail
When user needs to search a ticket	Option=6 Row=A	Print the detail that store in txt file named as "A5"	Passenger Ticket Information Row: A Seat: 5	<input checked="" type="checkbox"/> Pass <input type="checkbox"/> Fail

	Seat=5		Price: £200 Passenger's detail Name: virat Surname: kohli Email:virat@12.com	
When user input invalid row to search seat	Option= 1 Row=S Seat=3	Print a message to user that the input is invalid	Prints" Your selection is invalid. Please try again"	<input checked="" type="checkbox"/> Pass <input type="checkbox"/> Fail
When user input invalid seat number to search seat	Option= 1 Row=A Seat=30	Print a message to user that the input is invalid	Prints" Your selection is invalid. Please try again"	<input checked="" type="checkbox"/> Pass <input type="checkbox"/> Fail
When user need to see all the tickets deatail and Total price			Passenger Ticket Information Row: A Seat: 3 Price: £200 Passenger's detail Name: Sachin Surname: tendulkar Email:sachin@12.com Passenger Ticket Information Row: A Seat: 5 Price: £200 Passenger's detail Name: virat Surname: kohli Email:virat@12.com Total price is : £400	<input checked="" type="checkbox"/> Pass <input type="checkbox"/> Fail

Are there any specific parts of the coursework which you would like to get feedback?

1. I created this preview of the seating plan at the starting of this program for give a information about price of seats for user.

```
<<<<Welcome to Plane Management System!!>>>>

      1   2   3   4   5   6   7   8   9   10  11  12  13  14
A    0   0   0   0   0   0   0   0   0   0   0   0   0   0
B    0   0   0   0   0   0   0   0   0   0   0   0   0   0
C    0   0   0   0   0   0   0   0   0   0   0   0   0   0
D    0   0   0   0   0   0   0   0   0   0   0   0   0   0

Price of seats according to the column:
1-5 = £200
6-9 = £150
9-14= £180
```

2. I created a separate file called “Booking information” to store the details that entered by user.

```
public void save() {
    String fileName = "Booking Information"+File.separator +getRowLetter() +""+getSeatNo() + ".txt";
    try {
        String folderPath = "Booking Information"; // Folder path
        File folder = new File(folderPath);

        // Check if folder doesn't exist, then create it
        if (!folder.exists()) {
            folder.mkdirs();
        }
    }
}
```

You will need to demonstrate your understanding of the submitted code. Your tutor will arrange a coursework demonstration. During the coursework demonstration, your tutor will ask you to execute your program and questions on your code.

Failure to attend the demonstration will result in 0 for the coursework.

3) Code :

1.w2051998_PlaneManagement.java

```
import java.util.InputMismatchException;
import java.util.Scanner;
import java.io.File;

//plane management class manage the seating booking of the plane
public class w2051998_PlaneManagement {
    public static int[][] seats = { //2D array
        new int[14], new int[12], new int[12], new int[14] //This
        2D array used to make plane seat structure
    };

    public static void main(String[] args) {

        Tickets[][] tickets= new Tickets[4][14]; //This 2D array for
        ticket

        System.out.println("\n<<<<Welcome to Plane Management
        System!!>>>>");
        System.out.println("\n");
        System.out.println("
            1   2   3   4   5   6   7   8   9   10  11  12
13  14
      A   O   O   O   O   O   O   O   O   O   O   O   O
O   O
      B   O   O   O   O   O   O   O   O   O   O   O   O
      C   O   O   O   O   O   O   O   O   O   O   O   O
      D   O   O   O   O   O   O   O   O   O   O   O   O
O   O

            Price of seats according to the column:
            1-5 = £200
            6-9 = £150
            9-14= £180
            """);

        boolean terminate=false;
        do {
```

```

        int menuSelection = menu();
        switch (menuSelection) { //here i use switch case to select
the options
            case 0:
                terminate = true;
                System.out.println("Thank you for choosing
us.....Contact our team for more details"); //this for terminate the
whole program
                break;
            case 1:
                buy_seat(seats, tickets);
                break;
            case 2:
                cancel_seat(tickets);
                break;
            case 3:
                find_first_available();
                break;
            case 4:
                show_seating_plan();
                break;
            case 5:
                print_ticket_info(tickets);
                break;
            case 6:
                search_ticket(tickets);
                break;
            default:
                System.out.println("Invalid option. Please try
again."); //If user input invalid data this message will be thrown.
        }
    }while(!terminate);
    }

    private static int menu() { //this menu() prints menu option for
us
        Scanner scanner = new Scanner(System.in);
        int menuSelection;
        do {
            System.out.println("\n");
            System.out.println("*****
                                *
                                Menu Option
                                *
            *****

```

```

*****

        1) Buy a seat
        2) Cancel a seat
        3) Find first seat available
        4) Show seating plan
        5) Print tickets information and total sales
        6) Search tickets
        0) Quit

*****

        """);
        System.out.println("Select your option (0-6): ");
        while (!scanner.hasNextInt()) { // here check if input
is not an integer
            System.out.println("Incorrect input. Please enter a
number (0-6): ");
            scanner.next(); // Consume the invalid input
        }
        menuSelection = scanner.nextInt();
        if (menuSelection < 0 || menuSelection > 6) {
            System.out.println("Incorrect input. Please enter a
number between 0 and 6.");
        }
    } while (menuSelection < 0 || menuSelection > 6);
    return menuSelection;
}

//here this buy_seat option is used to book a seat and ask user
to input the other details that need to book a seat
    private static void buy_seat(int[][] seats, Tickets[][]
tickets){
        {Scanner scanner= new Scanner (System.in);
        try {
            //Prompt to ask user for row letter (A-D)
            System.out.println("Enter the row letter (A-D)");
            char rowLetter =
scanner.next().toUpperCase().charAt(0); //here we use char to get an
alphabet., and also convert in to uppercase
            int row = rowLetter - 'A'; /* Here we are subtract
A, Reason is in ascii A=65 so when we have an input row letter
A,B,C,D
            that shows 65 to array, but we created only 4 array so that
when we subtract the A we got A-A=0 in index 0 is the
            so we can access the 1st array */

            //prompt to ask user for seat number

```

```

        System.out.println("Enter the seat number : ");
        int seatNo = scanner.nextInt();

        scanner.nextLine();
        if (seats[row][seatNo - 1] == 0) { //this condition
checking whether seat is booked or not.

            seats[row][seatNo - 1] = 1;

            String name = "";
            while (name.trim().isEmpty()) {
                System.out.println("Enter passenger's name: ");
                name = scanner.nextLine();
                if (name.trim().isEmpty()) { //if user skip to
enter the name when book the ticket, program will not allow to next
step.

                    System.out.println("Passenger's Name cannot
be empty.");
                }
            }

            String surname = "";
            while (surname.trim().isEmpty()) {
                System.out.println("Enter passenger's surname:
");
                surname = scanner.nextLine();
                if (surname.trim().isEmpty()) { //if user skip to
enter the Surname when book the ticket, program will not allow to
next step.

                    System.out.println("Passenger's Surname
cannot be empty.");
                }
            }

            String email = "";
            while (email.trim().isEmpty()) {
                System.out.println("Enter passenger's Email id:
");
                email = scanner.nextLine();
                if (email.trim().isEmpty()) { //if user skip to
enter the email when book the ticket, program will not allow to book
ticket.

                    System.out.println("Email cannot be
empty.");
                }
            }
        }
    }
}

```

```

        Person newPerson = new
Person(name,surname,email);//here creating new object with user
details

        //Here calculating the price based on seat number
        int price;
        if (seatNo <= 5 ) {
            price= 200;
        } else if (seatNo <= 9) {
            price = 150;
        } else {
            price=180;
        }

        //Here create new Tickets object and store in
tickets array
        Tickets newTickets = new Tickets( rowLetter, seatNo
,price, newPerson);
        tickets[row][seatNo - 1] = newTickets;
        //to save ticket details
        newTickets.save();

        System.out.println("Your Ticket is booked. Thank
you...");
    } else {
        System.out.println("Sorry this ticket is already
booked...Try another seat to book");
    }
} catch
(InputMismatchException|ArrayIndexOutOfBoundsException e) {
    System.out.println("Your selection is invalid. Please
try again");
}

    }
}

private static void cancel_seat(Tickets[][] tickets) {
    {Scanner scanner= new Scanner (System.in);
    try {
        System.out.println("Enter the row letter (A-D)");
        char rowLetter = scanner.next().toUpperCase().charAt(0);
        int row = rowLetter - 'A';

        System.out.println("Enter the seat number : ");
        int seatNo = scanner.nextInt();
    }
}

```

```

        if (seats[row][seatNo - 1] == 1) { //this condition
checking whether seat is booked or not.

            seats[row][seatNo - 1] = 0;
            System.out.println("Your booking is cancelled");

            tickets[row][seatNo - 1] = null; //this used to remove
ticket from tickets array

            //here deleting the data which are entered when
buy_seat

            String fileName = rowLetter + "" + seatNo + ".txt";
            File delefile = new File(fileName);
            delefile.delete();

        } else {
            System.out.println("This seat is not booked yet");
        }
    } catch (InputMismatchException |
ArrayIndexOutOfBoundsException e) {
        System.out.println("Invalid input. Please try again");
    }
}

}

private static void find_first_available() {
    boolean available = false; //here this used to check if any
available seat is found
    for (int row = 0; row < seats.length; row++) { //here this
iterates over rows
        for (int seatNO = 0; seatNO < seats[row].length; seatNO++) { //here this it iterates over seats in each row
            if (seats[row][seatNO] == 0) {
                char rowLetter = (char) (row + 'A'); //here I
convert row index to corresponding letter (A-D)
                //Print first available seat information
                System.out.println("First available seats: row=
" + rowLetter + "\tSeat Number= " + (seatNO + 1) );
                available = true;
                break;
            }
        }
        if (available) { //Exit outer loop once an available seat
is found

```



```

        break;
    }
    if (!available){ // If no available seat is found after
iteration
        System.out.println("All Tickets are booked....");
    }
}

}

private static void show_seating_plan(){
    char rowLabel = 'A'; // Start with row label A
    for(int i = 0; i < seats.length; i++){
        System.out.print(rowLabel++ + ""); //print row label for
the row and move to next row
        for (int seat : seats[i]) { //here i used enhanced
loop
            if (seat == 0) {
                System.out.print(" 0 "); // Print '0' to
represent an available seat
            } else {
                System.out.print(" X "); // Print 'X' to
represent a booked seat
            }
        }
        System.out.println(); // Move to the next line after
printing seats for a row
    }
}

private static void print_ticket_info(Tickets[][] tickets){

    int totalprice = 0; //here initializing the total price
variable
    for(int row=0; row< tickets.length; row++) { // Iterate over
each row in the tickets array
        for (int seatNO = 0; seatNO < seats[row].length;
seatNO++) {
            if (tickets[row][seatNO] != null){
                tickets[row][seatNO].TicketDetails(); // Print
ticket details for the booked seat
                totalprice = totalprice +
tickets[row][seatNO].getPrice(); // Add the ticket price to the total
price
            }
        }
    }
}

```

```

        }
    }
    System.out.println("\n");
    System.out.println("Total price is : £" + totalprice); //here
    Printing the total price
}

private static void search_ticket(Tickets[][] tickets) {
    Scanner scanner = new Scanner(System.in);
    try {
        System.out.println("Enter the row letter (A-D)");
        char rowLetter = scanner.next().toUpperCase().charAt(0);
        int row = rowLetter - 'A';
        System.out.println("Enter the seat number : ");
        int seatNo = scanner.nextInt();

        if (tickets[row][seatNo - 1] != null) { // Check if a
            ticket is booked for the specified seat
            tickets[row][seatNo - 1].TicketDetails(); // Print
            ticket details for the booked seat
        } else {
            System.out.println("This seat is not booked. You can
            book this seat");
        }
    } catch (InputMismatchException |
    ArrayIndexOutOfBoundsException e) { // Catch exceptions for invalid
    input
        System.out.println("Invalid input. Please try again");
        scanner.nextLine();
    }
}
}
}

```

Person.java

```
public class Person {
    //attributes
    private String name;//here stores passenger's name

    private String surname;//here stores passenger's surname

    private String email;//here stores passenger's email

    //Constructor
    public Person(String name, String surname, String email){
        this.name= name;// Initialize name attribute
        this.surname= surname;// Initialize surname attribute
        this.email= email;// Initialize email attribute
    }

    //getters
    public String getName(){
        return this.name;//return passenger's name
    }

    public String getSurname(){
        return this.surname;//return passenger's surname
    }

    public String getEmail(){
        return this.email;//return passenger's email
    }

    //setter

    public void setName(String name){
        this.name = name;// Set the person's name
    }

    public void setSurname(String surname){
        this.surname = surname;
    }

    public void setEmail(String email){
        this.email = email;
    }
}
```

```

    }

    //print method
    public void personDetail(){//By this going to display person's
detail
        System.out.println( "\nPassenger's detail" +
            "\n\tName: " + getName() +
            "\n\tSurname: " + getSurname()+
            "\n\tEmail:" + getEmail()+ "\n");
    }
}

```

Tickets.java

```

import java.io.FileWriter;
import java.io.IOException;
import java.io.File;
public class Tickets {
    //attributes
    private char rowLetter;// Stores the row letter of the ticket

    private int seatNo;// Stores the seat number of the ticket

    private int price;// Stores the price of the ticket

    private Person person; // Stores the person associated with the
ticket

    //Constructor

    public Tickets(char rowLetter, int seat, int price, Person
person){

        this.rowLetter = rowLetter;// Initialize rowLetter attribute
        this.seatNo = seat;// Initialize seatNo attribute
        this.price = price;// Initialize price attribute
        this.person = person;// Initialize person attribute
    }

    public Tickets(char rowLetter, int seatNo, int price) {
    }

    //getters
    public char getRowLetter(){

```

```

        return this.rowLetter;
    } // Return the row letter of the ticket

    public int getSeatNo(){
        return this.seatNo;
    } // Return the seatNo of the ticket
    public int getPrice(){
        return this.price;
    } // Return the price of the ticket
    public Person getPerson(){
        return this.person;
    } // Return the person of the ticket
    //setters

    public void setRowLetter(char rowLetter) {
        this.rowLetter = rowLetter;
    } //set the row letter of the ticket

    public void setSeatNo(int seatNO) {
        this.seatNo = seatNO;
    } // set the seat number of the ticket

    public void setPrice(int price) {
        this.price = price;
    } // set the price of the ticket

    public void setPerson(Person person) {
        this.person = person;
    } //set the person of the ticket

    //Print method

    public void TicketDetails(){ //By this going to dispaly ticket
information
        System.out.println("Passenger Ticket Information " +
            "\n\tRow: " + getRowLetter()+
            "\n\tSeat: " + getSeatNo() +
            "\n\tPrice: £" + getPrice() );
        person.personDetail();
    }

    public void save() { //this method is used to save the ticket
information that entered by user
        String fileName = "Booking Information"+File.separator
+getRowLetter() +""+getSeatNo() + ".txt";//here I define the path
and file name
        try {

```

```

        String folderPath = "Booking Information";
        File folder = new File(folderPath);

        // here i check if folder doesn't exist, then create it
        if (!folder.exists()) {
            folder.mkdirs();
        }

        FileWriter file = new FileWriter(fileName); //Here i
created a FileWriter object
        file.write("-----\n");
        file.write("\tTicket information of seat booked\n");
        file.write("-----\n");
        file.write("\n");
        file.write("\tRow: " + getRowLetter() + "\n" + "\tSeat
Number: " + getSeatNo() + "\n");
        file.write("\tPrice: £" + getPrice() + "\n");
        file.write("\n-----Passenger's Information-----\n");
        file.write("\n");
        file.write("\tName of Passenger: " + person.getName() +
"\n");
        file.write("\tPassenger's Surname: " +
person.getSurname() + "\n");
        file.write("\tPassenger's E-mail Id: " +
person.getEmail() + "\n");

        file.close();// Close the FileWriter object

    } catch (IOException e) {
        System.out.println("Error occurred");
    }
}
}

```