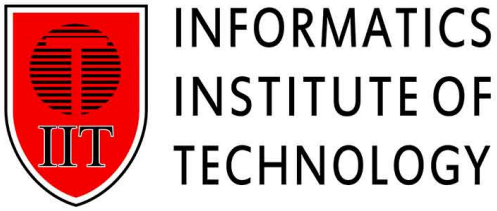

Software Development II

Coursework Report 2021/2022

Sandaru Bandara

W1866979

20200649

Task 01 – Source Code

```
package com.company;

import javax.swing.*;
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import java.nio.channels.ScatteringByteChannel;
import java.nio.charset.StandardCharsets;
import java.util.Arrays;
import java.util.InputMismatchException;
import java.util.Scanner;

public class Main {

    public static void main(String[] args) throws IOException {

        String[] cabins = new String[12];
        String[] BookedCabin = new String[12];

        initialise(cabins);
        boolean menuLoop = true;

        while (menuLoop) {

            System.out.println(".....Menu..... ");
            System.out.println("press 'A' to Add a customer to cabin ");
            System.out.println("press 'v' to View to all cabins ");
            System.out.println("press 'E' to View to empty cabins ");
            System.out.println("press 'D' to Delete customer from cabin ");
            System.out.println("press 'F' to Find cabin from customer name
");
            System.out.println("press 'S' to Store program data into file ");
            System.out.println("press 'L' to Load program data from file");
            System.out.println("press 'O' to View passengers Ordered
alphabetically by name ");
            System.out.println("press '0' to Exit the program ");
```

```
            System.out.println("......Select from the menu....");
            Scanner input2 = new Scanner(System.in);
            String input1 = input2.nextLine();

            switch (input1) {
                case "a", "A" -> addCabin(cabins);

                case "v", "V" -> viewCabin(cabins);

                case "e", "E" -> emptyCabins(cabins);

                case "d", "D" -> removecust(cabins);

                case "f", "F" -> findCabin(cabins);

                case "s", "S" -> storeProData(cabins);

                case "l", "L" -> lordProData();

                case "o", "O" -> booked(cabins, BookedCabin);

                case "0" -> menuLoop = false;

                default -> System.out.println("invalid input");
            }
        }
    }

    public static void initialise(String[] filledCabin) {
        for (int i = 0; i < filledCabin.length; i++) {
            filledCabin[i] = "e";
        }
    }


    public static void emptyCabins(String[] cabin) {
        System.out.println("Display empty cabins ....");
        for (int i = 0; i < cabin.length; i++) {
            if (cabin[i].equals("e")) {
                int cabNum = i + 1;
                System.out.println("Cabin " + cabNum + ",");
            }
        }
    }

    public static void addCabin(String[] cabin) {
        int cabNum = 0;
        String custName;

        while (cabNum < 13) {
            emptyCabins(cabin);
            Scanner input = new Scanner(System.in);
            try {
                System.out.println("select a cabin from empty cabins.. if you
want to exit program enter '0' ");
                cabNum = input.nextInt();
                cabNum = cabNum - 1;
```

```
                input.nextLine();
            } catch (InputMismatchException e) {
                System.out.println("Input is invalid Please try again ");
            }
            if (cabNum == -1) {
                break;
            } else if (cabNum < 12 && cabNum > -1) {
                System.out.println("please enter customer name...");
                custName = input.next();
                cabin[cabNum] = custName;
            }
        }
    }

    public static void viewCabin(String[] cabin) {
        int cabNum;
        for (int i = 0; i < cabin.length; i++) {
            if (cabin[i].equals("e")) {
                cabNum = i + 1;
                System.out.println("Cabin " + cabNum + " is empty...");
            } else {
                cabNum = i + 1;
                System.out.println("Cabin " + cabNum + " is booked by " +
cabin[i]);
            }
        }

    }

    public static void removecust(String[] cabin) {
        int cabNum = 0;
        String custName;
        while (cabNum < 12) {

            Scanner input = new Scanner(System.in);
            try {
                System.out.println("select the cabin number to remove the
customer or press 13 to exit");
                cabNum = input.nextInt();
                cabNum = cabNum - 1;

            } catch (InputMismatchException e) {
                System.out.println("invalid input please try again......");
            }
            if (cabNum == -1) {
                break;
            } else if (cabNum < 12 && cabNum > -1) {
                System.out.println("please enter customer name....");
                custName = input.next();
                if (cabin[cabNum].equals(custName)) {
                    cabin[cabNum] = "e";
                    cabNum = cabNum + 1;
                    System.out.println(custName + " is removed from cabin " +
cabNum);
                }
            }
        }
```

```
    }

    public static void BookedCab(String[] cabin) {
        int cabNum;
        for (int i = 0; i < cabin.length; i++) {
            if (!cabin[i].equals("e")) {
                cabNum = i + 1;
                System.out.println("cabin" + cabNum + " is booked by " +
cabin[i]);

            }
        }
    }

    public static void findCabin(String[] cabin) {
        String custName;

        Scanner input = new Scanner(System.in);
        System.out.println(" Enter customer name.....");
        custName = input.nextLine();

        for (int i = 0; i < cabin.length; i++) {
            if (cabin[i].equals(custName)) {
                int cabNum = i + 1;
                System.out.println(custName + " is in cabin  " + cabNum);
            }
        }
    }

    public static void storeProData(String[] cabin) throws IOException {

        try (FileWriter myWriter = new FileWriter("inputData.text")) {
            for (int i = 0; i < cabin.length; i++) {
                myWriter.write(cabin[i] + "\n");

            }
            myWriter.close();
            System.out.println("data write to th file.....");

        } catch (IOException e) {
            System.out.println("....error...");
            e.printStackTrace();
        }
    }

    public static void lordProData() throws IOException {
        int linecount = 1;
        try {
            File inputFile = new File("inputData.txt");
            Scanner read = new Scanner(inputFile);
            String fileline;
            while (read.hasNext()) {
                fileline = read.nextLine();
                if (!fileline.equals("e")) {
                    System.out.println("cabin " + linecount + "" + fileline);
                }
                linecount++;
```

```
            }
            read.close();
        } catch (IOException e) {
            System.out.println("Error IOException is " + e);
        }
    }

    public static void booked(String[] cabin, String[] booked) {
        initialise(booked);
        int inumber = 0;
        for (int i = 0; i < cabin.length; i++) {
            if (!(cabin[i].equals("e"))) {
                booked[inumber] = cabin[i];
                inumber++;
            }
        }
        System.out.println(Arrays.toString(cabin));
        for (int i = 0; i < booked.length; i++) ;
        {
            int i = 0;
            for (int j = i + 1; j < booked.length; j++) {
                if (!(cabin.equals("e"))) {
                    int cha = booked[i].compareTo(booked[j]);
                    if (cha > 0) {
                        String temp = booked[i];
                        booked[i] = booked[j];
                        booked[j] = temp;
                    }
                }
            }
            System.out.println("Ordered alphabetically");
            for (String customer : booked) {
                if (!customer.equals("e")) {
                    System.out.println(customer);

                }
            }
        }
    }
}
```

Task 02 – Main

```
package com.company;


import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileWriter;
import java.io.IOException;
```

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


public class main {
    static int numOfEmptyCabs = 12;

    public static void main(String[] args) {

        Passenger[] passengers = new Passenger[36];
        Cabin[] cabins = new Cabin[12];

        for (int i = 0; i < 12; i++) {
            cabins[i] = new Cabin();
        }
        for (int i = 0; i < passengers.length; i++) {
            passengers[i] = new Passenger();
        }
        initialise(cabins);

        boolean menuLoop = true;

        while (menuLoop) {

            System.out.println(".....Menu..... ");
            System.out.println("press 'A' to Add a customer to cabin ");
            System.out.println("press 'v' to View to all cabins ");
            System.out.println("press 'E' to View to empty cabins ");
            System.out.println("press 'D' to Delete customer from cabin ");
            System.out.println("press 'F' to Find cabin from customer name
");
            System.out.println("press 'S' to Store program data into file ");
            System.out.println("press 'L' to Load program data from file");
            System.out.println("press 'T' to View Expenses ");
            System.out.println("press '0' to Exit the program ");


            System.out.println("......Select from the menu....");
            Scanner input2 = new Scanner(System.in);
            String input1 = input2.nextLine();

            switch (input1) {
                case "a", "A" -> addPassenger(cabins,passengers);

                case "v", "V" -> displayCabin(cabins);

                case "e", "E" -> displayEmptyCabins(cabins);

                case "d", "D" -> removePassenger(cabins, passengers);

                case "f", "F" -> findCabinFromName(cabins);

                case "s", "S" -> storeData(cabins);

                case "l", "L" -> lordData(cabins);

                case "t", "T" -> printExpenses(passengers);
```

```
                case "0" -> menuLoop = false;

                default -> System.out.println("invalid input");

            }
        }
    }

    public static void initialise (Cabin[] array){
        for (int i = 0; i < array.length; i++) {
            array[i].setPassenger("e", "e", "e");
        }
    }

    public static void displayCabin(Cabin [] array) {
        for (int i = 0; i < array.length; i++) {
            int cabNum = i+1;
            System.out.println("Cabin " + cabNum + ",");
            array[i].displayCabin();
        }
    }

        public static void displayEmptyCabins (Cabin[] array){
            int emptycabins = 0;
            System.out.println("Display empty cabins ....");
            for (int i = 0; i < array.length; i++) {
                if (array [i].getPassenger01().equals("e") || (array
[i].getPassenger02().equals("e")|| (array
[i].getPassenger03().equals("e")))){
                    int cabNum = i + 1;
                    System.out.println("Cabin " + cabNum + ",");
                    emptycabins +=1;
                }
            }
            numOfEmptyCabs = emptycabins;

        }

        public static void addPassenger(Cabin[] array, Passenger [] array2) {
            int cabNum = 0;
            String customerFname;
            String customerLname;

            while (cabNum < 13) {
                displayEmptyCabins(array);              //display the cabins
                if (numOfEmptyCabs == 0) {
                    System.out.println(".....Cabins are filled.....");
                    break;
                }
                Scanner input = new Scanner(System.in);
                try {
                    System.out.println("select a cabin from empty cabins.. if
you want to exit program enter '0' ");
                    cabNum = input.nextInt();
                    cabNum = cabNum - 1;
                    input.nextLine();
```

```
                } catch (InputMismatchException e) {
                    System.out.println("Input is invalid Please try again ");
                    continue;
                }
                if (cabNum == -1) {
                    break;

                } else if (cabNum < 12 && cabNum > -1) {
                    if (array[cabNum].getPassenger01().equals("e") ||
(array[cabNum].getPassenger02().equals("e")) ||
(array[cabNum].getPassenger03().equals("e"))) {
                        System.out.println("Please Enter Your First Name...")
                        ;
                        customerFname = input.next();
                        System.out.println("Please Enter Your Sur Name...")
                        ;
                        customerLname = input.next();

                        //add passenger to object class.....

                        for (int i = 0; i < array2.length; i++) {

                            String name = array2[i].getName();
                            if (name.equals("e e")) {
                                array2[i].setNames(customerFname,
customerLname);

                                i = array2.length;
                            }
                        }
                        //Add Passenger To Cabin Object If Passenger Is 'e'
in Cabin Object
                        if (array[cabNum].passenger01.equals("e")) {
                            array[cabNum].passenger01 = customerFname + " " +
customerLname;
                        } else if (array[cabNum].passenger02.equals("e")) {
                            array[cabNum].passenger02 = customerFname + " " +
customerLname;
                        } else if (array[cabNum].passenger03.equals("e")) {
                            array[cabNum].passenger03 = customerFname + " " +
customerLname;
                        }
                        cabNum = cabNum + 1;
                        System.out.println(customerFname + " " +
customerLname + " added to cabin" + cabNum);
                    } else {
                        System.out.println("That cabin is no empty ");
                    }
                }
            else{
                    System.out.println(" invalid input please try again ");
                }
            }
        }

        public static void removePassenger(Cabin[] array,Passenger[] array2)
{
            String passengerName;
```

```
            boolean loop = true;

            while (loop) {
                int numberoferror = 0;
                Scanner input = new Scanner(System.in);
                System.out.println("enter passenger name need to remove or
press 0 to exit ");
                passengerName = input.nextLine();
                if (passengerName.equals("0")) {
                    loop = false;
                } else {
                    for (int i = 0; i < array.length; i++) {
//remove passenger
                        int cabNum = i + 1;
                        if
(array[i].passenger01.equalsIgnoreCase(passengerName)) {
                            array[i].removePassenger1();
                            System.out.println(passengerName + " removed form
Cabin" + cabNum);
                        } else if
(array[i].passenger02.equalsIgnoreCase(passengerName)) {
                            array[i].removePassenger2();
                            System.out.println(passengerName + " removed form
Cabin" + cabNum);
                        } else if
(array[i].passenger03.equalsIgnoreCase(passengerName)) {
                            array[i].removePassenger3();
                            System.out.println(passengerName + " removed form
Cabin" + cabNum);
                        } else {
                            numberoferror = numberoferror + 1;
                        }
                    }
                    if (numberoferror == array.length) {
                        System.out.println("name you entered is not in cabin
list");
                    }
                    for (int i = 0; i < array2.length; i++) {
                        String Name = array2[i].getName();
                        if (Name.equalsIgnoreCase(passengerName)) {
                            array2[i].removePassen();
                        }
                    }
                }
            }
        }

        public static void findCabinFromName(Cabin[] array){
            String passengerName;
            boolean loop = true;

            while (loop){
                Scanner input =new Scanner(System.in);
                System.out.println(" Enter passenger name to find cabin or
press '0' to exit");
                passengerName = input.nextLine();
                if (passengerName.equals("0")){
```

```
                    loop = false;
                }
                else {
                    int cabNum = 0;
                    for (int i = 0; i <array.length; i++){
                        if
(array[i].passenger01.equalsIgnoreCase(passengerName)||
(array[i].passenger02.equalsIgnoreCase(passengerName)
||(array[i].passenger03.equalsIgnoreCase(passengerName))))
                            cabNum = i +1;
                        System.out.println(passengerName+" Cabin is
"+cabNum);
                    }
                    if (cabNum == 0) {
                        System.out.println("name you entered is not in cabin
name list ");
                    }
                    else{
                        System.out.println(passengerName+" cabin is cabin
"+cabNum);
                    }
                }
            }
        }

    public static void storeData(Cabin[] array) {
//https://www.infoworld.com/article/2076301/learn-how-to-store-data-in-
objects.html
        try {
            FileWriter dataWrite = new FileWriter("inputData.text");
            for (int i= 0; i <array.length; i++){
                dataWrite.write(array[i].getPassenger());

            }
            dataWrite.close();
            System.out.println(".....Data stored.....");
        }
         catch( IOException e) {
             System.out.println("....Error....");
             e.printStackTrace();
         }
    }
    public static void lordData(Cabin[] array) {
        String[] data = new String[36];

        try {
            File myFile = new File("inputData.text");
            Scanner reader = new Scanner(myFile);
            int n = 0;
            while (reader.hasNext()) {
                data[n] = reader.nextLine();
                n++;

            }
            int count = 0;
            int cabNum = 0;
            while (count < 36) {
```

```
                for (int i = 0; i < array.length; i++) {
                    String data01 = data[count];
                    count += 1;
                    String data02 = data[count];
                    count += 1;
                    String data03 = data[count];
                    count += 1;
                    cabNum = i + 1;
                    System.out.println("\n Cabin " + cabNum + "\n Passenger
01 " + data01 + "\n Passenger 02 " + data02 + "\n Passenger 03 " + data03);
                }
            }
            System.out.println("\n ......Data Lorded ......");
        } catch (FileNotFoundException e) {
            System.out.println(".....Error.....");
            e.printStackTrace();
        }
    }
    public static void printExpenses(Passenger[] array){
        for (int i = 0; i < array.length;i++){
            int expense = array[i].getExpenses();
            String name = array[i].getName();
            if (!name.equals("e e")){
                System.out.println(name+" Total Expenses " +expense);
            }

        }
        int totalExpense = array[0].getTotalExpenses();
        System.out.println("Total Expenses "+totalExpense);

    }
}
```

cabin class

```
package com.company;

public class Cabin {
    String passenger01;
    String passenger02;
    String passenger03;

    public void displayCabin(){
        System.out.println(" Passenger 1 "+passenger01);
        System.out.println(" Passenger 2 "+passenger02);
        System.out.println(" Passenger 3 "+passenger03);
    }
    public String getPassenger(){ return
passenger01+"\n"+passenger02+"\n"+passenger03+"\n";}

    public void setPassenger(String passenger01,String passenger02,String
passenger03){
        this.passenger01 = passenger01;
        this.passenger02 = passenger02;
        this.passenger03 = passenger03;
```

```
    }
    public String getPassenger01(){
        return passenger01;
    }
    public String getPassenger02() {

        return passenger02;
    }
    public String getPassenger03(){

        return passenger03;
    }

    public void setPassenger01(String name){passenger01 = name;}
    public void setPassenger02(String name){passenger02 = name;}
    public void setPassenger03(String name){passenger03 = name;}

    public void removePassenger1(){passenger01 = "e" ;}
    public void removePassenger2(){passenger02 = "e"; }
    public void removePassenger3(){passenger03 = "e"; }


}
```

passenger class

```
package com.company;

public class Passenger {
    private String firstName = "e";
    private String surName = "e";
    private int expenses = 100;
    private static int totalExpenses;

    public void Name(String fname,String sname){
        firstName = fname;
        surName = sname;
        totalExpenses = totalExpenses + expenses;

    }
    public String getName() { return firstName+" "+surName;}
    public int getExpenses() { return expenses;}
    public void expenses(int expenses){
        this.expenses = expenses;
        totalExpenses = totalExpenses + expenses;

    }

    public int getTotalExpenses(){return totalExpenses;}
    public void removePassen(){
        firstName = "e";
        surName = "e";
        totalExpenses = totalExpenses + expenses;
    }
```

```
    public void setNames(String customerFname, String customerLname) {

    }
}
```

Task 03 – Queue

```
package com.company;


import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileWriter;
import java.io.IOException;
import java.util.InputMismatchException;
import java.util.Scanner;

public class Queue {

    static int start = 0;
    static int end = 0;
    static int numOfEmptyCabins = 12;

    static String[] passengerQueue = new String[15];

    public static void main(String[] args) {
        Passenger[] passengers = new Passenger[36];
        Cabin[] cabins = new Cabin[12];

        for (int i = 0; i < 12; i++) {
            cabins[1] = new Cabin();

        }
        for (int i = 0; i < passengers.length; i++) {
            passengers[i] = new Passenger();
        }

        initialise(cabins);
        initialiseQueue(passengerQueue);

        boolean menuloop = true;

        while (menuloop) {
            setNumOfEmptycabs(cabins);          // getting number of empty
cabins

            System.out.println(".....Menu..... ");
            System.out.println("press 'A' to Add a customer to cabin ");
            System.out.println("press 'v' to View to all cabins ");
            System.out.println("press 'E' to View to empty cabins ");
            System.out.println("press 'D' to Delete customer from cabin ");
            System.out.println("press 'F' to Find cabin from customer name
");
            System.out.println("press 'S' to Store program data into file ");
            System.out.println("press 'L' to Load program data from file");
            System.out.println("press 'T' to View Expenses ");
            System.out.println("press '0' to Exit the program ");


            System.out.println("......Select from the menu....");
```

```
            Scanner input2 = new Scanner(System.in);
            String input1 = input2.nextLine();

            switch (input1) {
                case "a", "A" -> {
                        if (passengerQueueCount() == passengerQueue.length &&
numOfEmptyCabins > 0) {
                            addPassenger(cabins, passengers);
                        }
                        else{
                        queueManage(cabins);
                    }
                }

                case "v", "V" -> displayCabin(cabins);

                case "e", "E" -> displayEmptyCabins(cabins);

                case "d", "D" -> removePassenger(cabins, passengers);

                case "f", "F" -> findCabinFromName(cabins);

                case "s", "S" -> storeData(cabins);

                case "l", "L" -> lordData(cabins);

                case "t", "T" -> printExpenses(passengers);

                case "0" -> System.exit(0);

                default -> System.out.println("invalid input");

            }
        }
    }

    private static void setNumOfEmptycabs(Cabin[] array) {
        int emptyCabins = 0;
        for (int i = 0; i < array.length; i++) {
            if (array[i].getPassenger01().equals("e") ||
array[i].getPassenger02().equals("e") ||
array[i].getPassenger03().equals("e")) {
                emptyCabins += 1;
            }
        }
        numOfEmptyCabins = emptyCabins;
    }

    public static void initialise(Cabin[] array) {
        for (int i = 0; i < array.length; i++) {
            array[i].setPassenger("e", "e", "e");
        }
    }

    public static void displayCabin(Cabin[] array) {
        for (int i = 0; i < array.length; i++) {
            int cabNum = +1;
```

```
            System.out.println("Cabin " + cabNum + ",");
            array[i].displayCabin();
        }
    }
    // Display th empty cabins if passengers assigning  with 'e'


    public static void displayEmptyCabins(Cabin[] array) {
        int emptyCabCount = 0;
        System.out.println("Display empty cabins ....");
        for (int i = 0; i < array.length; i++) {
            if (array[i].getPassenger01().equals("e") ||
(array[i].getPassenger02().equals("e") ||
(array[i].getPassenger03().equals("e")))) {
                int cabNum = i + 1;
                System.out.println("Cabin " + cabNum + ",");
                emptyCabCount = emptyCabCount + 1;
            }
        }
        numOfEmptyCabins = emptyCabCount;

    }

    public static void addPassenger(Cabin[] array, Passenger[] array2) {
        int cabNum = 0;
        String customerFname;
        String customerLname;

        while (cabNum < 13) {
            displayEmptyCabins(array);              //Display the cabins
            if (numOfEmptyCabins == 0) {
                System.out.println(".....Cabins are filled.....");
                break;
            }
            Scanner input = new Scanner(System.in);
            try {
                System.out.println("select a cabin from empty cabins.. if you
want to exit program enter '0' ");
                cabNum = input.nextInt();
                cabNum = cabNum - 1;
                input.nextLine();
            } catch (InputMismatchException e) {
                System.out.println("Input is invalid Please try again ");
                continue;
            }
            if (cabNum == -1) {
                break;

            } else if (cabNum < 12 && cabNum > -1) {
                if (array[cabNum].getPassenger01().equals("e") ||
(array[cabNum].getPassenger02().equals("e")) ||
(array[cabNum].getPassenger03().equals("e"))) {
                    System.out.println("Please Enter Your First Name...")
                    ;
                    customerFname = input.next();
                    System.out.println("Please Enter Your Sur Name...")
                    ;
```

```
                    customerLname = input.next();

                    //add passenger to object class.....

                    for (int i = 0; i < array2.length; i++) {

                        String name = array2[i].getName();
                        if (name.equals("e e")) {
                            array2[i].setNames(customerFname, customerLname);
                            i = array2.length;
                        }
                    }
                    //Add Passenger To Cabin Object If Passenger Is 'e' in
Cabin Object
                    if (array[cabNum].passenger01.equals("e")) {
                        array[cabNum].passenger01 = customerFname + " " +
customerLname;
                    } else if (array[cabNum].passenger02.equals("e")) {
                        array[cabNum].passenger02 = customerFname + " " +
customerLname;
                    } else if (array[cabNum].passenger03.equals("e")) {
                        array[cabNum].passenger03 = customerFname + " " +
customerLname;
                    }
                    cabNum = cabNum + 1;
                    System.out.println(customerFname + " " + customerLname +
" added to cabin" + cabNum);
                } else {
                    System.out.println("That cabin is no empty ");
                }
            } else {
                System.out.println(" invalid input please try again ");
            }
        }
    }



    public static void removePassenger(Cabin[] array, Passenger[] array2) {
        String passengerName;
        boolean loop = true;

        while (loop) {
            int numberoferror = 0;
            Scanner input = new Scanner(System.in);
            System.out.println("enter passenger name need to remove or press
0 to exit ");
            passengerName = input.nextLine();
            if (passengerName.equals("0")) {
                loop = false;
            } else {
                for (int i = 0; i < array.length; i++) {
//remove passenger
                    int cabNum = i + 1;
                    if (array[i].passenger01.equalsIgnoreCase(passengerName))
{
                        array[i].removePassenger1();
```

```
                        System.out.println(passengerName + " removed form
Cabin" + cabNum);
                    } else if
(array[i].passenger02.equalsIgnoreCase(passengerName)) {
                        array[i].removePassenger2();
                        System.out.println(passengerName + " removed form
Cabin" + cabNum);
                    } else if
(array[i].passenger03.equalsIgnoreCase(passengerName)) {
                        array[i].removePassenger3();
                        System.out.println(passengerName + " removed form
Cabin" + cabNum);
                    } else {
                        numberoferror = numberoferror + 1;
                    }
                }
                if (numberoferror == array.length) {
                    System.out.println("name you entered is not in cabin
list");
                }
                for (int i = 0; i < array2.length; i++) {
                    String Name = array2[i].getName();
                    if (Name.equalsIgnoreCase(passengerName)) {
                        array2[i].removePassen();
                    }
                }
            }
        }
    }

    public static void findCabinFromName(Cabin[] array) {
        String passengerName;
        boolean loop = true;

        while (loop) {
            Scanner input = new Scanner(System.in);
            System.out.println(" Enter passenger name to find cabin or press
'0' to exit");
            passengerName = input.nextLine();
            int cabNum = 0;
            if (passengerName.equals("0")) {
                loop = false;
            } else {
                cabNum = 0;
                for (int i = 0; i < array.length; i++) {
                    if (array[i].passenger01.equalsIgnoreCase(passengerName)
|| (array[i].passenger02.equalsIgnoreCase(passengerName) ||
(array[i].passenger03.equalsIgnoreCase(passengerName))))
                        cabNum = i + 1;
                }
                System.out.println(passengerName + " Cabin is " + cabNum);
            }
            if (cabNum == 0) {
                System.out.println("name you entered is not in cabin name
list ");
            }
        }
```

```
    }

    // Write data to file

    public static void storeData(Cabin[] array) {
//https://www.infoworld.com/article/2076301/learn-how-to-store-data-in-
objects.html
        try {
            FileWriter dataWrite = new FileWriter("inputData.txt");
            for (int i=0; i<array.length;i++){
                dataWrite.write(array[i].getPassenger());

            }
            dataWrite.close();
            System.out.println(".....Data stored.....");
        } catch (IOException e) {
            System.out.println("....Error....");
            e.printStackTrace();
        }
    }
    //read from the text file

    public static void lordData(Cabin[] array) {
        String[] data = new String[36];

        try {
            File myFile = new File("inputData,txt");
            Scanner reader = new Scanner(myFile);
            int n = 0;
            while (reader.hasNext()) {
                data[n] = reader.nextLine();
                n++;

            }
            int count = 0;
            int cabNum = 0;
            while (count < 36) {
                for (int i = 0; i < array.length; i++) {
                    String data01 = data[count];
                    count += 1;
                    String data02 = data[count];
                    count += 1;
                    String data03 = data[count];
                    count += 1;
                    cabNum = i + 1;
                    System.out.println("\n Cabin " + cabNum + "\n Passenger
01 " + data01 + "\n Passenger 02 " + data02 + "\n Passenger 03 " + data03);
                }
            }
            System.out.println("\n ......Data Lorded ......");
        } catch (FileNotFoundException e) {
            System.out.println(".....Error.....");
            e.printStackTrace();
        }
    }

    public static void printExpenses(Passenger[] array) {
```

```
        for (int i = 0; i < array.length; i++) {
            int expense = array[i].getExpenses();
            String name = array[i].getName();
            if (!name.equals("e e")) {
                System.out.println(name + " Total Expenses " + expense);
            }

        }
        int totalExpense = array[0].getTotalExpenses();
        System.out.println("Total Expenses " + totalExpense);

    }

    public static void initialiseQueue(String[] array) {
        for (int i = 0; i < array.length; i++) {
            array[i] = "e";

        }
    }

    public static void addQueue(String name) {
        for (int i = 0; i < passengerQueue.length; i++) {
            if (passengerQueue[i].equals("e")) {
                end = i;
                i = passengerQueue.length;
            }
        }
        passengerQueue[end] = name;
        System.out.println(name + " Added to Queue");
        end++;

    }

    public static String takeQueue(Passenger[] array2, String inputName) {
        String passengerName = "";
        if (end > start) {
            System.out.println("\n .....input taken....." +
passengerQueue[start]);
            passengerName = passengerQueue[start];

            for (int i = 0; i < passengerQueue.length; i++) {
                if (i != passengerQueue.length - 1) {
                    passengerQueue[i] = passengerQueue[i + 1];
                    passengerQueue[i + 1] = "e";
                }
            }

            String[] split =
passengerName.split("");//https://www.geeksforgeeks.org/split-string-java-
examples/
            String customerFname = split[0];
            String customerlname = split[1];

            for (int i = 0; i < array2.length; i++) {
                String name = array2[i].getName();
                if (name.equals(inputName)) {
                    array2[i].removePassen();
```

```
                    array2[i].setNames(customerFname, customerlname);
                    i = array2.length;
                }
            }
        } else {
            System.out.println("........Queue is Empty........");}
            return passengerName;
        }

    public static void queueManage(Cabin[] array){
        boolean loop = true;
        while (loop){
            System.out.println("Cabins are filld with passengers added to th
queue...");
            Scanner input = new Scanner(System.in);
            System.out.println("enter your first name or Press '0' to exit");
            Scanner customerFname = input.reset();
            if (customerFname.equals("0")){
                break;
            }
            System.out.println("enter your sur name or Press '0' to exit");
            Scanner customerLname = input.reset();
            if (customerLname.equals("0")){
                break;
            }
            String name = customerFname+ " "+customerLname;

            if (numOfEmptyCabins == 0){
                addQueue(name);
            }
        }
    }
    public static int passengerQueueCount(){
        int passengerQueueCount = 0;
        for (int i = 0; i <passengerQueue.length; i++){
            if (passengerQueue[i].equals("e")){
                passengerQueueCount = passengerQueueCount+1;
            }
        }
        return passengerQueueCount;
    }
}
```

Task 04 – Testing

Array solution

Test Case	Expected Result	Actual Result	Pass/Fail
(Cabins Initialized correctly) After the program starts, Press 'E'	Display "all empty cabins"	Display "all empty cabins"	Pass
(Add customer "Bob" to cabin 5) Select A, enter "Bob"	Press 'v' Displays "Cabin 5 is booked by bob" for cabin5	Displays "Cabin 5 is booked by bob" for cabin 5	pass
(view all cabins) Select V, Display all cabins	Display all cabins, Display "Cabin 5 is booked by bob"	Display all cabins, "Cabin 5 is booked by bob"	pass
Select E, display empty cabins	Display empty cabins	Display empty cabins	pass
Select S, store data to the file	store data to the file	store data to the file	pass
Select L, lord data from file	Display "cabin 5 bob"	Display "cabin 5 bob	pass
(Delete customer) Press D, enter cabin number, enter customer name	Display "bob is removed from cabin 5"	Display "bob is removed from cabin 5"	pass
(alphabetic order) Press O,	Display "ordered alphabetically" display the name in order	Display "ordered alphabetically" display the name in order	pass
(find cabin from the name) Pres F, enter customer name	Display "bob is in cabin 5"	Display "bob is in cabin 5"	Pass
(Exit the program) Press 0,	Exit the program	Exit the program	pass

Class solution

Test Case	Expected Result	Actual Result	Pass/Fail
(Cabins Initialized correctly) After the program starts, Press 'V'	Displays 'Cabin passengers empty' for all cabins	Displays 'Cabin passengers empty' for all cabins	Pass
(Add customer "sandaru bandara " to cabin 5) Select A, enter "sandaru","bandara"	Press 'v' Displays sandaru bandara added to cabin5" for cabin5	Displays "sandaru bandara added to cabin5" for cabin 5	pass
(Display expenses) Press T	Display the expenses and total exencess	Display the expenses and total exencess	pass
Select E, display all empty cabins	Display all empty cabins (passengers are not in cabin)	Display all empty cabins (passengers are not in cabin)	pass
Select S, store data to the file	store data to the file	store data to the file	pass
Select L, lord data from file	Display the lord data	Display the lord data	pass
(Find cabin from name) Pess F, enter name	Display "sandaru bandara Cabin is 5"	Display "sandaru bandara Cabin is 5"	pass
(Exit the program) Press 0,	Exit the program	Exit the program	pass

Task 04 – Testing – Discussion

First used my own data to check the program. Then used the given data in the table. First, I checked the arrays are initialized. Then checked the menu function and menu. And create an object array and test that. then test the adding function. I used the given data to test if the name added to the array ran the display all cabins function. Then tested the display empty cabin's function. On the delete customer/ passenger function I deleted the name entered earlier. To check whether the customer deleted I used the display of all cabin functions. To test the find cabin function added another customer to a cabin and ran the find cabin function and enter the same name. And got the correct output. The store data added some customers to the cabins and run the store data function and opened the text file to check if the data are written correctly After that ran the load data function and it displayed all the customer is in a cabin in an array solution. And in the cabin solution, it printed all the cabins. The class solution to get the expenses ran the function and printed the passenger from the passenger object array and showed each passenger's expenses and the total expense of all the passengers. The total expenses add and removed when the passenger is added passenger object array or when the set names function to run and when removing a passenger from the passenger object array. To test the queue function filled all the cabins with passengers and added extra passengers to the queue and remove passengers from the cabin to check if the passenger from the queue were added. To check the expenses change happened and the show expense's function, For the real-world program the class solution is more suitable,

Self-Evaluation form

Criteria	Component marks	Expected Mark
Task 1 One mark for each option (A,V,E,D,F,S,L,O) Menu works correctly	24 6	**24** **6**
Student comment: fully implemented and working.		
Task 2 Cabin class correctly implemented. Passenger class correctly implemented. Expenses correctly reported.	14 10 6	**14** **0** **6**
Student comment: fully implemented and working hard.		
Task 3 Waiting list queue implementation "A: Add"works correctly "D: Delete"works correctly Circular queue implementation	10 3 3 4	**10** **3** **3** **3**
Student comment: fully implement and working hard		
Task 4 Test case coverage and reasons Writeup on which version is better and why	6 4	**6** **3**
Student comment: test all the parts correctly and all parts are working.		
Coding Style (Comments, indentation, style) Complete the self-evaluation form indicating what you have accomplished to ensure appropriate feedback.	7 3	**6** **2**
Student comment: put comments and links I used to create the program.		
Totals		**(100)**
Demo: At the discretion of your tutor, you may be called on to give a demo of your work to demonstrate understanding of your solutions. If you cannot explain your code and are unable to point to a reference within your code of where this code was found (i.e., in a textbook or on the internet) then significant marks will be lost for that marking component. If you do not attend a requested demo your mark will be capped at 50%.		

References

https://www.infoworld.com/article/2076301/learn-how-to-store-data-in-objects.html

https://www.geeksforgeeks.org/split-string-java-examples/

https://www.w3schools.com/java/java_arrays.asp

https://www.w3schools.com/java/java_classes.asp

https://www.geeksforgeeks.org/string-in-switch-case-in-java/