Informatics Institute of Technology

Department of Computing

Software Development II Coursework Report

Module	: 4COSC010C.3: Software Development II
Module Leader	: Deshan Sumanathilake (Program)/ Sulari Fernando (Design)
Date of submission	: 08/08/2022
Student ID	: 20210243 / w1912867
Student First Name	: Himashi
Student Surname	: Kodithuwakku

"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 : Himashi Kodithuwakku

Student ID : 20210243

Test Cases – Array version

	Test Case	Expected Result	Actual Result	Pass/Fail
1	Fuel Queue Initialized Correctly After program starts, 100 or VFQ	Displays 'Empty' for all queues.	Displays 'Empty' for all Queues.	Pass
2	Add passenger "Jane" to Queue 2 102 or ACQ Enter Queue: 2 Enter token number (0-5):0 Enter Name: Jane	Display 'Jane added to the queue 2 successfully"	Display "Customer added to the queue 2" The stock reduce by 10 liters	Pass
3	View all empty queues 101 or VEQ	Display all the "empty" queues Except queue 2 , index number 0.	Display all "Empty" queues Except queue 2 index number 0 as displayed as "token number 0 occupied by Jane"	Pass
4	Remove a customer from a queue 103 or AFQ Enter queue number :2 Enter token number (0-5):0	Remove a customer from a specific location Display "customer 0 is Removed"	Display "customer 0 is Removed" The stock added 10 liters back	Pass
5	Remove a served customer 104 or PCQ	Remove served customer	Remove served customer	Pass
6	View customers sorted in alphabetical order 105 or VCS Enter queue number :2	View customers in alphabetical order	Display "the names in alphabetical order are:"	Pass

7	Store program data into a file 106 or SPD	Display “wrote in to a file” Store data into a text file	Display ”Successfully wrote to the file.”	Pass
8	Load program data from a file 107 or LPD	Loaded data from a text file	Display”Loaded data from file”	Pass
9	View remaining fuel stock 108 or AFS	Displays the remaining stock which was calculated when adding a customer .	Display “Remaining Fuel stock 6590 l”	Pass
10	Add fuel stock	Displays the stock served fuel	Display” Served fuel stock : 10 l”	Pass
11	Exit the program	End the program Display” Exit the program”	Display” Exit the program”	Pass

Test Cases – Class version

	Test Case	Expected Result	Actual Result	Pass/Fail
12	Fuel Queue Initialized Correctly After program starts, 100 or VFQ	Displays ‘null’ for all queues.	Displays ‘null’ for all Queues.	Pass
13	Add passenger “Jane” to Queue 2 102 or ACQ Enter Queue number (1-5): 2 Enter token number (0-5): 0 Enter First name: Jane Enter Last name : Leblanc Enter Vehicle number :CHA-3235 Enter number of liters required:20	Display ‘Jane added to the queue 2 successfully”	Display “Jane added to t queue 2”	Pass

14	View all empty queues 101 or VEQ	Display all the “empty” queues Except queue 2 , index number 0.	Display all ”Empty” queues Except queue 2 index number 0 as displayed as “token number 0 occupied by [Jane,Leblnac,CHA-3235,20]” The stock reduce by 20 liters	Pass
15	Remove a customer from a queue 103 or AFQ Enter queue number :2 Enter token number (0-5):0	Remove a customer from a specific location Display “customer 0 is Removed”	Display “customer 0 is Removed” The stock added 10 liters back	Pass
16	Remove a served customer 104 or PCQ	Displays the first customer was removed from the queue 2 and the next customer added into empty space.	Displays the first customer was removed from the queue 2 and the next customer added into empty space. The 1st customer was removed the queue 2 index 0 and all the other customers in the behind is shifted to the in front.	Pass
17	View customers sorted in alphabetical order 105 or VCS Enter queue number :2	All the customers should display in alphabetical order	Display “the names in alphabetical order are:”	Pass

18	Store program data into a file 106 or SPD	Display “wrote in to a file” Store data into a text file	Display ”Successfully wrote to the file.”	Pass
19	Load program data from a file 107 or LPD	Loaded data from a text file	Display”Loaded data from file”	Pass
20	View remaining fuel stock 108 or AFS	Displays the remaining stock which was calculated when adding a customer .	Display “Remaining Fuel stock 6590 l”	Pass
21	Add fuel stock	Displays the stock served fuel	Display” Served fuel stock : 10 l”	Pass
22	Income of each fuel queue Enter Queue number (1-5): 2	Display income value of each queues	Displays “queue 2 – income :Rs 8600”	Pass
23	Exit the program	End the program Display” Exit the program”	Display” Exit the program”	Pass

Fuel Queue Management System-Array version (Discussion)

I designed a program for fuel center. This fuel center has 3 pumps where maximum 6 customers can be hold simultaneously in a queue. There are 10 tasks in a menu. Fuel center will have exactly 6600 liters in their stock. First task is display all queues. Second task displays all the empty places in 3 queues. When a customer added to queue, system has to input customer name and queue number, token number. After added the customer in to a queue 10 liters reduce from the stock. This program assume 10 liters served for each customer. When Stock reaches a 500 liters, it will display warning message. In add fuel stock task display total of liters served for customers. As well as, in remaining stock task display total of remaining stock value. When customer remove from a queue in program, it will require queue number and token number. Another task is each queue's customer names are display in alphabetical order. this program has to store customer data in to a text file and display load data from stored text file .when user enter EXT or 999, program will exit.

Fuel Queue Management System-class version

In class version fuel center has 5 pumps. Class version has Menu as 11 tasks. Many tasks work same as array version .When customer add in to the queue program will ask customer first name, last name, vehicle number, number of required liters and queue number token number. There is additional task in this class version. It called as income of each fuel queue. This task count income price of each queue/ price of one liters as Rs. 430.

Fuel Queue Management System.

Code : Array version

```
package cw;



import java.util.Scanner;

import java.io.File;

import java.io.FileOutputStream;



import java.io.IOException;



public class arrayversion {



    static String CustomerName;

    static int queue_1 = 6;

    static int queue_2 = 6;

    static int queue_3 = 6;



    static int fuelStock = 6600;

    static int stockReachvalue = 500;
    static int served= 0;



    //creating three  arrays for 3 queues

    static String[] queue1 = new String[6];

    static String[] queue2 = new String[6];

    static String[] queue3 = new String[6];

    static Scanner chooseMethod = new Scanner(System.in);



    public static void mainMenu() {


        Scanner chooseMethod = new Scanner(System.in);


        //initialize

        initialize(queue1, queue2, queue3);


        //Main menu


        while (true) {
```

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

System.out.println("Choose from here");

System.out.println(" ");

System.out.println("100 or VFQ -> View all Fuel Queues");

System.out.println("101 or VEQ -> View all Empty Queue ");

System.out.println("102 or ACQ -> Add customer to a queue ");

System.out.println("103 or RCQ -> Remove Customer from a Queue ");

System.out.println("104 or PCQ -> Remove a served customer ");

System.out.println("105 or VCS -> View Customer Sorted in alphabetical order ");

System.out.println("106 or SPD -> Store program data into file ");

System.out.println("107 or LPD -> Load program data from file ");

System.out.println("108 or STK -> View Remaining Fuel Stock ");

System.out.println("109 or AFS -> Add Fuel Stock ");

System.out.println("999 or EXT -> Exit the program");

System.out.println(" ");

System.out.print("Enter the menu number or code -> ");

String choose = chooseMethod.next();

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

```
// getting input



if ((choose.equals("100") || (choose.equalsIgnoreCase("VCQ")))) {
    System.out.println("View all Fuel Queues");
    viewAllQueues();
} else if ((choose.equals("101") || (choose.equalsIgnoreCase("VEQ")))) {
    System.out.println("View all empty  Queues");
    displayEmptyQueues();

} else if ((choose.equals("102") || (choose.equalsIgnoreCase("ACQ")))) {
    System.out.println(" Add customer to a queue");
    addToQueue();

} else if ((choose.equals("103") || (choose.equalsIgnoreCase("RCQ")))) {
    System.out.println("Remove Customer from a Queue");
    removeCustomer();
```

```
} else if ((choose.equals("104") || (choose.equalsIgnoreCase("PCQ")))) {

    removeServedCustomer();

} else if ((choose.equals("105") || (choose.equalsIgnoreCase("VCS")))) {
    System.out.println("View Customer Sorted in alphabetical order");
    alphabetOrderName();

} else if ((choose.equals("106") || (choose.equalsIgnoreCase("SPD")))) {
    System.out.println("Store program data into file");
    storeData();


} else if ((choose.equals("107") || (choose.equalsIgnoreCase("LPD")))) {
    System.out.println("Load program data from file");
    loadData();

} else if ((choose.equals("108") || (choose.equalsIgnoreCase("STK")))) {
    System.out.println("View Remaining Fuel Stock");
    remainingStock();
```

```
}

            } else if ((choose.equals("109") || (choose.equalsIgnoreCase("AFS")))) {

                System.out.println("Add Fuel Stock");

                addFuelStock();



            } else if ((choose.equals("999") || (choose.equalsIgnoreCase("EXT")))) {

                {

                    System.out.println("Exit the program");

                    break;


                }



            }



        }

    }
```

```
private static void initialize(String[] que1ref, String[] que2ref, String[] que3ref) {
    // creating empty places
    for (int i = 0; i < queue_1; i++) que1ref[i] = "Empty";
    System.out.println("initialise");


    for (int i = 0; i < queue_2; i++) que2ref[i] = "Empty";




    for (int i = 0; i < queue_3; i++) que3ref[i] = "Empty";


}


public static void viewAllQueues() {
    //viewing all the queues
    System.out.println("Queue 1\n");


    for (int i = 0; i < queue_1; ) {
        System.out.println(queue1[i]);
        i++;
}

System.out.println();

System.out.println("Queue 2\n");



for (int i = 0; i < queue_2; ) {

    System.out.println(queue2[i]);

    i++;



}



System.out.println(" ");

System.out.println("Queue 3\n");



for (int i = 0; i < queue_3; ) {

    System.out.println(queue3[i]);

    i++;



}
```

```
    }



    public static void addToQueue() {

        // add customer into one of the three queue when  customer added they are given  token
number to customer.


        System.out.println("Enter queue number (1-3) :");

        int queunum = chooseMethod.nextInt();



        if (queunum == 1) {



            System.out.println("Enter  token number (0-5) :");

            int token = chooseMethod.nextInt();


            System.out.println("Enter  customer name :");

            CustomerName = chooseMethod.next();
    queue1[token] = CustomerName;

    System.out.println("Customer added to the queue 1");

    // reduct 10 liters from the stock

    fuelStock = fuelStock - 10;

    // add 10 liters to served stock

    served = served +10;





} else if (queunum == 2) {



    System.out.println("Enter  token number (0-5) :");

    int token = chooseMethod.nextInt();



    System.out.println("Enter  customer name :");

    CustomerName = chooseMethod.next();

    queue2[token] = CustomerName;

    System.out.println("Customer added to the queue 2");

    // reduct 10 liters from the stock

    fuelStock = fuelStock - 10;
    // add 10 liters to served stock
    served = served +10;


} else if (queunum == 3) {

    System.out.println("Enter   token number (0-5) :");
    int token = chooseMethod.nextInt();

    System.out.println("Enter  customer name :");
    CustomerName = chooseMethod.next();
    queue3[token] = CustomerName;
    System.out.println("Customer added to the queue 3");
    // reduct 10 liters from the stock
    fuelStock = fuelStock - 10;
    // add 10 liters to served stock
    served = served +10;


}
```

```
}


public static void displayEmptyQueues() {

    // display all the empty places in the queues

    System.out.println("  ");
    System.out.println("Queue 1\n");
    for (int i = 0; i < queue_1; i++) {
        if (queue1[i].equals("Empty")) {
            System.out.println("Empty");
        } else
            System.out.println(" token number" + i + " occupied by " + queue1[i]);
    }
    System.out.println("  ");
    System.out.println("Queue 2\n");
    for (int i = 0; i < queue_2; i++) {
        if (queue2[i].equals("Empty")) {
            System.out.println("Empty");
        } else

            System.out.println(" token number" + i + " occupied by " + queue2[i]);

    }

    System.out.println("  ");

    System.out.println("Queue 3\n");

    for (int i = 0; i < queue_3; i++) {

        if (queue3[i].equals("Empty")) {

            System.out.println("Empty");

        } else

            System.out.println(" token number" + i + " occupied by " + queue3[i]);

    }

}


public static void removeCustomer() {


    // remove a customer  from specific location


    int tokenNo;

    System.out.println("Enter queue number :");
```

```
int queunum = chooseMethod.nextInt();

if (queunum == 1) {
    System.out.println("Enter  token number (0-5) :");
    tokenNo = chooseMethod.nextInt();
    for (int i = 0; i < queue_1; i++) {
        if ((i) == tokenNo) {
            queue1[i] = "Empty";
            // when customer remove from the queue ,after that 10 liters add to fuel stock
            fuelStock = fuelStock + 10;
            System.out.println("customer " + " " + i + " " + "is Removed");
        }
    }

} else if (queunum == 2) {

    System.out.println("Enter  token number (0-5):");
    tokenNo = chooseMethod.nextInt();
    for (int i = 0; i < queue_2; i++) {
        if ((i) == tokenNo) {
}
            queue2[i] = "Empty";
            fuelStock = fuelStock + 10;
            System.out.println("customer " + " " + i + " " + "is Removed");
        }
    }
} else {
    System.out.println("Enter  token number (0-5):");
    tokenNo = chooseMethod.nextInt();
    for (int i = 0; i < queue_3; i++) {
        if ((i) == tokenNo) {
            queue3[i] = "Empty";
            fuelStock = fuelStock + 10;
            System.out.println("customer " + " " + i + " " + "is Removed");
        }else{
            System.out.println("");
        }

    }

}
```

```
}


public static void removeServedCustomer() {


    // remove  customer after received 10 liters


    System.out.println("Enter queue number:");
    int  queuenum = chooseMethod.nextInt();



     if (queuenum == 1) {
         queue1[0] ="Empty";


         for (int i = 0;i <5;i++){
             queue1[i] =queue1[i+1];
         }
         queue1[5] ="Empty";


     }
```

```
if (queuenum == 2) {

    queue2[0] ="Empty";



    for (int i = 0;i <5;i++){

        queue2[i] =queue2[i+1];

    }

    queue2[5] ="Empty";



}



if (queuenum == 3) {

    queue3[0] ="Empty";



    for (int i = 0;i <5;i++){

        queue3[i] =queue3[i+1];

    }

    queue3[5] ="Empty";

}
```

```
}

    public static void alphabetOrderName () {

    // display customers name according to alphabetical order

        System.out.println("Enter queue number:");
        int queuenum = chooseMethod.nextInt();

        if (queuenum == 1) {

            // create string array called names
            String temp;
            for (int i = 0; i < queue_1; i++) {
                for (int j = i + 1; j < queue_1; j++) {

                    // to compare one string with other strings
                    if (queue1[i].compareTo(queue1[j]) > 0) {
                        // swapping
                        temp = queue1[i];
                        queue1[i] = queue1[j];
                        queue1[j] = temp;
                    }
                }
            }

            // print output array
            System.out.println(
                    "The names in alphabetical order are: ");
            for (int i = 0; i < queue_1; i++) {
                System.out.println(queue1[i]);
            }

        } else if (queuenum == 2) {
            String temp;
            for (int i = 0; i < queue_2; i++) {
                for (int j = i + 1; j < queue_2; j++) {
                    // to compare one string with other strings
                    if (queue2[i].compareTo(queue2[j]) > 0) {
                        // swapping
                        temp = queue2[i];
                        queue2[i] = queue2[j];
                        queue2[j] = temp;
                    }
                }
            }

            // print output array
            System.out.println(
                    "The names in alphabetical order are: ");
            for (int i = 0; i < queue_2; i++) {
                System.out.println(queue2[i]);
            }

        } else {
            String temp;
            for (int i = 0; i < queue_3; i++) {
                for (int j = i + 1; j < queue_3; j++) {

                    // to compare one string with other strings
                    if (queue3[i].compareTo(queue3[j]) > 0) {
                        // swapping
                        temp = queue3[i];
                        queue3[i] = queue3[j];
                        queue3[j] = temp;
                    }
                }
            }

            // print output array
            System.out.println(
                    "The names in alphabetical order are: ");
            for (int i = 0; i < queue_3; i++) {
                System.out.println(queue3[i]);
            }
        }
```

```
}


public static void storeData () {

// store data into a file

    try {

        FileOutputStream storefile = new FileOutputStream("storedata.txt");



        for (int i = 0; i < queue_1; i++) {

            String filedata;



            if (queue1[i].equals("empty")) {

                filedata = "token " + " " + i + " " + "is empty\n";

            } else

                filedata = "token " + " " + i + " occupied by " + queue1[i] + "\n";

            storefile.write(filedata.getBytes());

        }



        for (int i = 0; i < queue_2; i++) {

            String filedata;
        if (queue2[i].equals("empty")) {
            filedata = "token " + " " + i + " " + "is empty\n";
        } else
            filedata = "token " + " " + i + " occupied by " + queue2[i] + "\n";
        storefile.write(filedata.getBytes());
    }

    for (int i = 0; i < queue_3; i++) {
        String filedata;

        if (queue3[i].equals("empty")) {
            filedata = "token " + " " + i + " " + "is empty\n";
        } else
            filedata = "token " + " " + i + " occupied by " + queue3[i] + "\n";
        storefile.write(filedata.getBytes());
    }

    System.out.println("Successfully wrote to the file.");
    storefile.close();
} catch (IOException e) {
        System.out.println("An error occurred.");
    }
}

public static void loadData () {
// load data from  stored file
    try {
        File inputFile = new File("storedata.txt");
        Scanner readfile = new Scanner(inputFile);
        String filedata;
        while (readfile.hasNextLine()) {
            filedata = readfile.nextLine();
            System.out.println(filedata);
        }
        readfile.close();
    } catch (IOException e) {
        System.out.println("Error");
    }
```

```
}


public static void remainingStock () {

//display remaining  fuel stock


   System.out.println("Remaining Fuel stock " + fuelStock +" l");




   if (fuelStock <= stockReachvalue) {


      System.out.print("Warning!");


   }


}


public static void   addFuelStock(){

// display served fuel stock


System.out.println("Served fuel stock :" + served+" l");
```

```
    }




    public static void main (String[]args){

    //main menu


        mainMenu();

    }



}
```

```
    }
```

Fuel Queue Management System. (class version)

Code : – Passenger class

```
package coursework;


public class Passenger {
    private String firstName;
    private String lastName;
    private String vehicleNo;
    private static  String noOfLiters;


    public Passenger() {

    }
```

```
//getter for the first name
public  String getFirstName() {
    return firstName;
}
//setter for the first name
public void setFirstName(String firstName){
    this.firstName = firstName;
}



//getter for the last name
public  String getLastName() {
    return lastName;
}
//setter for the last name
public void setLastName(String lastName){
    this.lastName = lastName;
}


//getter to get the vehicle number
public String getVehicleNumber() {
    return vehicleNo;
}


//setter to set  the vehicle number
public void setVehicleNumber(String vehicleNo) {
    this.vehicleNo = vehicleNo;
}


//getter to get the number of liters
public static String getNoOfLiters() {
    return noOfLiters;
}
```

```
    //setter to set  the number of liters

    public void setNoOfLiters(String  noOfLiters) {

        this.noOfLiters = noOfLiters;

    }









}
```

Code : – FuelQueue class

```
package coursework;



import java.io.File;

import java.io.FileOutputStream;

import java.io.IOException;

import java.util.Scanner;
import java.util.Arrays;

public class FuelQueue {


    static int queue = 6;


    //creating 5  arrays for 5 queues
    static String[][] queue1 = new String[queue][];
    static String[][] queue2 = new String[queue][];
    static String[][] queue3 = new String[queue][];
    static String[][] queue4 = new String[queue][];
    static String[][] queue5 = new String[queue][];


    static int fuelStock = 6600;
    static int stockReachvalue = 500;
    static int served = 0;

    static int liters = 0;


    static  int  income_1 = 0;
    static  int  income_2 = 0;
    static  int  income_3 = 0;
    static  int  income_4 = 0;
    static  int  income_5 = 0;

    static int queue1_liters =0;
    static int queue2_liters =0;
    static int queue3_liters =0;
    static int queue4_liters =0;
    static int queue5_liters =0;
    static Scanner chooseMethod = new Scanner(System.in);

    public static void mainMenu() {

        Scanner chooseMethod = new Scanner(System.in);



        //Main menu

        while (true) {
            System.out.println("====|Fuel Queue Management System|==== ");
            System.out.println(" ");
            System.out.println("Choose from here");
            System.out.println(" ");
            System.out.println("100 or VFQ -> View all Fuel Queues");
    System.out.println("101 or VEQ -> View all Empty Queue ");

    System.out.println("102 or ACQ -> Add customer to a queue ");

    System.out.println("103 or RCQ -> Remove Customer from a Queue ");

    System.out.println("104 or PCQ -> Remove a served customer ");

    System.out.println("105 or VCS -> View Customer Sorted in alphabetical order ");

    System.out.println("106 or SPD -> Store program data into file ");

    System.out.println("107 or LPD -> Load program data from file ");

    System.out.println("108 or STK -> View Remaining Fuel Stock ");

    System.out.println("109 or AFS -> Add Fuel Stock ");

    System.out.println("110 or IFQ -> Income of each Fuel Queue ");

    System.out.println("999 or EXT -> Exit the program");

    System.out.println(" ");

    System.out.print("Enter the menu number or code -> ");

    String choose = chooseMethod.next();

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

```
// getting input



if ((choose.equals("100") || (choose.equalsIgnoreCase("VCQ")))) {
    System.out.println("View all Fuel Queues");
    viewAllQueues();

} else if ((choose.equals("101") || (choose.equalsIgnoreCase("VEQ")))) {
    System.out.println("View all empty  Queues");
     displayEmptyQueues();

} else if ((choose.equals("102") || (choose.equalsIgnoreCase("ACQ")))) {
    System.out.println(" Add customer to a queue");
    AddPassenger();

} else if ((choose.equals("103") || (choose.equalsIgnoreCase("RCQ")))) {
    System.out.println("Remove Customer from a Queue");
     removeCustomer();
```

```
} else if ((choose.equals("104") || (choose.equalsIgnoreCase("PCQ")))) {

    removeServedCustomer(queue1,queue2,queue3,queue4,queue5);

} else if ((choose.equals("105") || (choose.equalsIgnoreCase("VCS")))) {
   System.out.println("View Customer Sorted in alphabetical order");
    alphabetOrderName();

} else if ((choose.equals("106") || (choose.equalsIgnoreCase("SPD")))) {
   System.out.println("Store program data into file");
      storeData();



} else if ((choose.equals("107") || (choose.equalsIgnoreCase("LPD")))) {
   System.out.println("Load program data from file");
    loadData();
```

```
} else if ((choose.equals("108") || (choose.equalsIgnoreCase("STK")))) {
    System.out.println("View Remaining Fuel Stock");
      remainingStock();


} else if ((choose.equals("109") || (choose.equalsIgnoreCase("AFS")))) {
    System.out.println("Add Fuel Stock");
      addFuelStock();


} else if ((choose.equals("110") || (choose.equalsIgnoreCase("IFQ")))) {
    System.out.println("Income of each Fuel Queue");
    incomeFuelQueue();




} else if ((choose.equals("999") || (choose.equalsIgnoreCase("EXT")))) {
    {
        System.out.println("Exit the program");
        break;
```

```
            }

        }



    }

}





public static int AddPassenger() {

    // get customer details



    Passenger person = new Passenger();

    System.out.println("Enter Queue number (1-5):");

    int queuenum = chooseMethod.nextInt();

    chooseMethod.nextLine();

    System.out.println("Enter token number (0-5):");

    int tokennum = chooseMethod.nextInt();
```

```
System.out.println("Enter First name:");
person.setFirstName(chooseMethod.next());
System.out.println("Enter Last name:");
person.setLastName(chooseMethod.next());
System.out.println("Enter Vehicle number:");
person.setVehicleNumber(chooseMethod.next());
System.out.println("Enter number of liters required:");
person.setNoOfLiters(chooseMethod.next());

System.out.println(person.getFirstName() + " added to queue" + queuenum);

// store customer data in array
String[] details = {person.getFirstName(), person.getLastName(), person.getVehicleNumber(), person.getNoOfLiters()};
liters = Integer.parseInt(details[3]);
fuelStock = fuelStock - liters;
served = served + liters;
```

```
for (int i =0 ;i < queue; i ++)

    if (queuenum == 1) {



        queue1[tokennum] = details;

        queue1_liters = Integer.parseInt(details[3]);





    } else if (queuenum == 2) {

        queue2[tokennum] = details;

        queue2_liters = Integer.parseInt(details[3]);





    } else if (queuenum == 3) {

        queue3[tokennum] = details;

        queue3_liters = Integer.parseInt(details[3]);
        } else if (queuenum == 4) {

            queue4[tokennum] = details;

            queue4_liters = Integer.parseInt(details[3]);



        } else if (queuenum == 5) {

            queue5[tokennum] = details;

            queue5_liters = Integer.parseInt(details[3]);



        }


return fuelStock;
```

```
}

public static void viewAllQueues() {

    //viewing all the queues

    System.out.println("Queue 1\n");

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

      System.out.println( Arrays.toString(queue1[i]));

  }

    System.out.println();

    System.out.println("Queue 2\n");

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

        System.out.println( Arrays.toString(queue2[i]));

    }

    System.out.println();

    System.out.println("Queue 3\n");

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

        System.out.println( Arrays.toString(queue3[i]));

    }
    System.out.println();

    System.out.println("Queue 4\n");

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

        System.out.println( Arrays.toString(queue4[i]));

    }

    System.out.println();

    System.out.println("Queue 5\n");

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

        System.out.println( Arrays.toString(queue5[i]));

    }




}



public static void displayEmptyQueues() {



    // display all the empty places in the queues
```

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

System.out.println("Queue 1\n");

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

    if (queue1[i]== null) {

        System.out.println("Empty");

    } else

        System.out.println(" token number " + i + " occupied by " + Arrays.toString(queue1[i]));

}

System.out.println("  ");

System.out.println("Queue 2\n");

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

    if (queue2[i]== null) {

        System.out.println("Empty");



    } else

        System.out.println(" token number " + i + " occupied by " + Arrays.toString(queue2[i]));
    }

    System.out.println("  ");

    System.out.println("Queue 3\n");

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

      if (queue3[i]== null) {

        System.out.println("Empty");

      } else

        System.out.println(" token number " + i + " occupied by "
+Arrays.toString(queue3[i])) ;

    }

    System.out.println("  ");

    System.out.println("Queue 4\n");

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

      if (queue4[i]==null) {

        System.out.println("Empty");

      } else

        System.out.println(" token number " + i + " occupied by "
+Arrays.toString(queue4[i]));

    }
```

```
        System.out.println("  ");
        System.out.println("Queue 5\n");
        for (int i = 0; i < queue; i++) {
            if (queue5[i]==null) {
                System.out.println("Empty");
            } else
                System.out.println(" token number " + i + " occupied by " + Arrays.toString(queue5[i]));
        }
    }


    public static void removeCustomer() {


        // remove a customer  from specific location


        int tokenNo;
        System.out.println("Enter queue number :");
        int queunum = chooseMethod.nextInt();
```

```
if (queunum == 1) {
    System.out.println("Enter  token number :");
    tokenNo = chooseMethod.nextInt();
    for (int i = 0; i < queue; i++) {
        if ((i) == tokenNo) {
            queue1[i] = null;
            // when customer remove from the queue , liters add into fuel stock
            fuelStock = fuelStock +   queue1_liters ;
            System.out.println("customer " + " " + i + " " + "is Removed");
        }
    }

} else if (queunum == 2) {


    System.out.println("Enter  token number :");
    tokenNo = chooseMethod.nextInt();
    for (int i = 0; i < queue; i++) {
        if ((i) == tokenNo) {

            queue2[i] = null;

            fuelStock = fuelStock +   queue2_liters ;

            System.out.println("customer " + " " + i + " " + "is Removed");

        }

    }

} else if (queunum ==3) {

    System.out.println("Enter  token number :");

    tokenNo = chooseMethod.nextInt();

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

        if ((i) == tokenNo) {

            queue3[i] = null;

            fuelStock = fuelStock +   queue3_liters ;

            System.out.println("customer " + " " + i + " " + "is Removed");

        }else{

            System.out.println("");

        }
    }

}else if (queunum ==4) {

    System.out.println("Enter  token number :");
    tokenNo = chooseMethod.nextInt();
    for (int i = 0; i < queue; i++) {
        if ((i) == tokenNo) {
            queue4[i] = null;
            fuelStock = fuelStock +   queue4_liters ;
            System.out.println("customer " + " " + i + " " + "is Removed");
        } else {
            System.out.println("");
        }
    }

}else{
```

```
            System.out.println("Enter  token number :");

            tokenNo = chooseMethod.nextInt();

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

                if ((i) == tokenNo) {

                    queue5[i] = null;

                    fuelStock = fuelStock +   queue5_liters ;

                    System.out.println("customer " + " " + i + " " + "is Removed");

                } else {

                    System.out.println("");

                }

            }

        }

    }



    public static void removeServedCustomer(String [][]
queue1,String[][]queue2,String [][] queue3,String [][]queue4,String[][] queue5 ) {



        // remove  served customer
```

```
System.out.println("Enter queue number:");

int  queuenum = chooseMethod.nextInt();




if (queuenum == 1) {



    queue1[0] =null;



    for (int i = 0;i <5;i++){

        queue1[i] =queue1[i+1];

    }

    queue1[5] =null;



}



if (queuenum == 2) {
    queue2[0] =null;


    for (int i = 0;i <5;i++){
        queue2[i] =queue2[i+1];
    }
    queue2[5] =null;


}


if (queuenum == 3) {


    queue3[0] =null;


    for (int i = 0;i <5;i++){
        queue3[i] =queue3[i+1];
    }
    queue3[5] =null;
}
```

```
if (queuenum == 4) {


    queue4[0] =null;


    for (int i = 0;i <5;i++){
        queue4[i] =queue4[i+1];
    }
    queue4[5] =null;
}


if (queuenum == 5) {


    queue5[0] =null;


    for (int i = 0;i <5;i++){
        queue5[i] =queue5[i+1];
    }
```

```
        queue5[5] =null;

    }







}

public static void alphabetOrderName () {


    // display customers name according to alphabetical order


    System.out.println("Enter queue number:");

    int queuenum = chooseMethod.nextInt();


    if (queuenum == 1) {


        // create string array called names

        String []temp;
        for (int i = 0; i < queue; i++) {
            for (int j = i + 1; j < queue; j++) {

                // to compare one string with other strings
                if  (Arrays.toString(queue1[i]).compareTo(Arrays.toString(queue1[j]))
> 0) {
                    // swapping
                    temp = queue1[i];
                    queue1[i] = queue1[j];
                    queue1[j] = temp;
                }
            }
        }

        // print output array
        System.out.println(
                "The names in alphabetical order are: ");
        for (int i = 0; i < queue; i++) {
                System.out.println(Arrays.toString(queue1[i]));
            }


        } else if (queuenum == 2) {
            String[] temp;
            for (int i = 0; i < queue; i++) {
                for (int j = i + 1; j < queue; j++) {

                    // to compare one string with other strings
                    if  (Arrays.toString(queue2[i]).compareTo(Arrays.toString(queue2[j])) > 0) {
                        // swapping
                        temp = queue2[i];
                        queue2[i] = queue2[j];
                        queue2[j] = temp;
                    }
                }
            }
```

```
      // print output array
      System.out.println(
          "The names in alphabetical order are: ");
      for (int i = 0; i < queue; i++) {
        System.out.println(Arrays.toString(queue2[i]));
      }

    } else if (queuenum == 3) {
      String [] temp;
      for (int i = 0; i < queue; i++) {
        for (int j = i + 1; j < queue; j++) {

          // to compare one string with other strings
          if  (Arrays.toString(queue3[i]).compareTo(Arrays.toString(queue3[j]))
> 0) {
            // swapping
            temp = queue3[i];
```

```
                queue3[i] = queue3[j];
                queue3[j] = temp;
            }
        }
    }

    // print output array
    System.out.println("The names in alphabetical order are: ");
    for (int i = 0; i < queue; i++) {
        System.out.println(Arrays.toString(queue3[i]));
    }
} else if (queuenum == 4) {
    String[] temp;
    for (int i = 0; i < queue; i++) {
        for (int j = i + 1; j < queue; j++) {

            // to compare one string with other strings
```

```
                    if  (Arrays.toString(queue4[i]).compareTo(Arrays.toString(queue4[j]))
> 0) {
                        // swapping
                        temp = queue4[i];
                        queue4[i] = queue4[j];
                        queue4[j] = temp;
                    }
                }
            }


            // print output array
            System.out.println(
                    "The names in alphabetical order are: ");
            for (int i = 0; i < queue; i++) {
                System.out.println(Arrays.toString(queue4[i]));
            }
        } else if (queuenum == 5) {
            String []temp;
        for (int i = 0; i < queue; i++) {
            for (int j = i + 1; j < queue; j++) {


                // to compare one string with other strings
                if  (Arrays.toString(queue5[i]).compareTo(Arrays.toString(queue5[j]))
> 0) {
                    // swapping
                    temp = queue5[i];
                    queue5[i] = queue5[j];
                    queue5[j] = temp;
                }
            }
        }


        // print output array
        System.out.println(
                "The names in alphabetical order are: ");
        for (int i = 0; i < queue; i++) {
            System.out.println(Arrays.toString(queue5[i]));
        }
    }


}

public static void storeData () {
    // store  customer data into a text file
    try {
        FileOutputStream storefile = new FileOutputStream("storedata.txt");


        for (int i = 0; i < queue; i++) {
            String filedata;

            if (queue1[i] ==(null)) {
                filedata = "token " + " " + i + " " + "is empty\n";
            } else
                filedata = "token " + " " + i + " occupied by " + Arrays.toString(queue1[i]) + "\n";
            storefile.write(filedata.getBytes());
        }



        for (int i = 0; i < queue; i++) {
            String filedata;

            if (queue2[i]==(null)) {
                filedata = "token " + " " + i + " " + "is empty\n";
            } else
                filedata = "token " + " " + i + " occupied by " + Arrays.toString(queue2[i]) + "\n";
            storefile.write(filedata.getBytes());
        }
```

```
    for (int i = 0; i < queue; i++) {
        String filedata;


        if (queue3[i]==(null)) {
            filedata = "token " + " " + i + " " + "is empty\n";
        } else
            filedata = "token " + " " + i + " occupied by " +Arrays.toString(queue3[i]) + "\n";
        storefile.write(filedata.getBytes());
    }



    for (int i = 0; i < queue; i++) {
        String filedata;


        if (queue4[i]==(null)) {
            filedata = "token " + " " + i + " " + "is empty\n";
```

```
                } else
                    filedata = "token " + " " + i + " occupied by " +
Arrays.toString(queue4[i]) + "\n";
                storefile.write(filedata.getBytes());
            }




            for (int i = 0; i < queue; i++) {
                String filedata;



                if (queue5[i]==(null)) {
                    filedata = "token " + " " + i + " " + "is empty\n";
                } else
                    filedata = "token " + " " + i + " occupied by " +
Arrays.toString(queue5[i]) + "\n";
                storefile.write(filedata.getBytes());
            }
```

```
        System.out.println("Successfully wrote to the file.");
        storefile.close();
    } catch (IOException e) {
        System.out.println("An error occurred.");
    }
}

public static void loadData () {
    // load data from  stored file
    try {
        File inputFile = new File("storedata.txt");
        Scanner readfile = new Scanner(inputFile);
        String filedata;
        while (readfile.hasNextLine()) {
            filedata = readfile.nextLine();
            System.out.println(filedata);
        }
        readfile.close();
    } catch (IOException e) {
        System.out.println("Error");
    }



}
public static void remainingStock () {
    //display remaining  fuel stock

    System.out.println("Remaining Fuel stock " + fuelStock +" l");



    if (fuelStock <= stockReachvalue) {

        System.out.print("Warning!");

    }
```

```
}


public static void   addFuelStock(){
    // display served fuel stock


    System.out.println("Served fuel stock :" + served+" l");


}


public static void incomeFuelQueue(){
    int q1, q2, q3,q4,q5;




    System.out.println("Enter  Queue number (1-5) \n ");
    int queuenum = chooseMethod.nextInt();
    if (queuenum == 1){
```

```
    income_1 = income_1 +queue1_liters;

    q1 =income_1 * 430;

    System.out.println("Queue 1 - income : Rs "+ q1 +"\n");



}else if (queuenum == 2){



    income_2 = income_2 +queue2_liters;

    q2 =income_2 * 430;

    System.out.println("Queue 2 - income : Rs "+ q2 +"\n");



}else if(queuenum == 3){

    income_3 = income_3 +queue3_liters;

    q3 =queue3_liters * 430;

    System.out.println("Queue 3 - income : Rs "+ q3 +"\n");



}else if(queuenum == 4){

    income_4 = income_4 +queue4_liters;

    q4 =queue4_liters * 430;
```

```
}

        System.out.println("Queue 4 - income : Rs "+ q4 +"\n");


    }else if(queuenum == 5){
        income_5 = income_5 +queue5_liters;
        q5 =queue5_liters * 430;
        System.out.println("Queue 5 - income : Rs "+ q5 +"\n");
    }



    }




public static void main (String[]args){
    //main menu


    mainMenu();
}
```

```
}
```

<<END>>