



INFORMATICS
INSTITUTE OF
TECHNOLOGY

UNIVERSITY OF
WESTMINSTER

Software Development II

4COSC010C.2

Individual Course Work **Report**

- Payagalawaduge Kaveesh Gayethra Sirican Fernando
- Uow Number - w1809865
- IIT Number – 20200142

INTRODUCTION

This program about hotel program which contains array version and class version. A standard program was laid out for an inn with eight rooms. I think it was important to make this app as simple and detailed as possible so it will be user-friendly. It started out easy and developed into something far more complicated. It will first show the client the most menu options, and then it will show the client the least menu options.

It will guide the client to the menu choices that are most appealing to them based on their preferences.

Test cases

Student Name: Kaveesh Fernando			Student ID:w1809865	
TEST PLAN for array version				
Submit completed test plan with your code solution				
Test No.	Test Input	Expected Result	Actual Result (or state 'not attempted')	Pass / Fail ('Actual Result' matches 'Expected Result')
1	<p>press A and room number: 0, room name: kaveesh, first name: kaveesh, surname: fernando, credit card number: 1234, number of guest: 2</p> <p>Add room number 1, room name: anupa, first name: anupa, surname: rathnayaka, Credict card number:4567 Nuber of guests:3</p> <p>Add room number: 2, Romm name: akila, first name: akila, surname:srikantha, Credict card number: 5678, number of guests: 4</p> <p>After adding press v</p>	<p>Displays room 0 occupied by kaveesh kaveesh fernando 1234 2</p> <p>room 1 occupied by anupa anupa rathnayaka 4567 3</p> <p>room 2 occupied by akila akila srikantha 5678 4</p>	<p>room 0 occupied by kaveesh kaveesh fernando 1234 2</p> <p>room 1 occupied by anupa anupa rathnayaka 4567 3</p> <p>room 2 occupied by akila akila srikantha 5678 4</p>	Pass

2	After adding press E	Display room 3 is empty room 4 is empty room 5 is empty room 6 is empty room 7 is empty	Display room 3 is empty room 4 is empty room 5 is empty room 6 is empty room 7 is empty	Pass
3	Then press D and enter the room number 1 for delete	Display Room Deleted	Display Room Deleted	Pass
4	Press F and enter the name kaveesh for find	Displays Name matched with the customer	Display Name matched with the customer	Pass
5	Press V	room 0 occupied by kaveesh kaveesh fernando 1234 2 room 2 occupied by akila akila srikantha 5678 4	room 0 occupied by kaveesh kaveesh fernando 1234 2 room 2 occupied by akila akila srikantha 5678 4	Pass
6	press S for store data and press L to load files.	Name and Room number is: kaveesh at: 0 First name is: kaveesh Surname is: fernando Credit card number is: 1234 Number of guests is: 2 Name and Room number is: akila at: 2 First name is: akila Surname is: srikantha Credit card number is: 5678 Number of guests is: 4	Name and Room number is: kaveesh at: 0 First name is: kaveesh Surname is: fernando Credit card number is: 1234 Number of guests is: 2 Name and Room number is: akila at: 2 First name is: akila Surname is: srikantha Credit card number is: 5678	Pass

			Number of guests is: 4	
7	Then press q to get alphabetic order	kaveesh akila	kaveesh akila	Pass

Student Name: Kaveesh Fernando	Student ID:w1809865
---------------------------------------	---------------------

TEST PLAN for class version				
Submit completed test plan with your code solution				

Test No.	Test Input	Expected Result	Actual Result (or state 'not attempted')	Pass / Fail ('Actual Result' matches 'Expected Result')
1	<p>press A and room number: 0, room name: kaveesh, first name: kaveesh, surname: fernando, credit card number: 1234, number of guest: 2</p> <p>Add room number 1, room name: anupa, first name: anupa, surname: rathnayaka, Credit card number:4567 Nuber of guests:3</p> <p>Add room number: 2, Romm name: akila, first name: akila, surname:srikantha, Credit card number: 5678, number of guests: 4</p> <p>After adding press v</p>	<p>Displays room 0 occupied by kaveesh kaveesh fernando 1234 2</p> <p>room 1 occupied by anupa anupa rathnayaka 4567 3</p> <p>room 2 occupied by akila akila srikantha 5678 4</p>	<p>room 0 occupied by kaveesh kaveesh fernando 1234 2</p> <p>room 1 occupied by anupa anupa rathnayaka 4567 3</p> <p>room 2 occupied by akila akila srikantha 5678 4</p>	Pass
2	After adding press E	<p>Display room 3 is empty room 4 is empty</p>	<p>Display room 3 is empty room 4 is empty</p>	Pass

		room 5 is empty room 6 is empty room 7 is empty	room 5 is empty room 6 is empty room 7 is empty	
3	Then press D and enter the room number 1 for delete	Display Room Deleted	Diply Room Deleted	Pass
4	Press F and enter the name kaveesh for find	Displays Name matched with the customer	Display Name matched with the customer	Pass
5	Press V	room 0 occupied by kaveesh kaveesh fernando 1234 2 room 2 occupied by akila akila srikantha 5678 4	room 0 occupied by kaveesh kaveesh fernando 1234 2 room 2 occupied by akila akila srikantha 5678 4	Pass
6	press S for store data and press L to load files.	Name and Room number is: kaveesh at: 0 First name is: kaveesh Surname is: fernando Credit card number is: 1234 Number of guests is: 2 Name and Room number is: akila at: 2 First name is: akila Surname is: srikantha Credit card number is: 5678 Number of guests is: 4	Name and Room number is: kaveesh at: 0 First name is: kaveesh Surname is: fernando Credit card number is: 1234 Number of guests is: 2 Name and Room number is: akila at: 2 First name is: akila Surname is: srikantha Credit card number is: 5678 Number of guests is: 4	Pass
7	Then press q to get alphabetic order	kaveesh akila	kaveesh akila	Pass

1.Array version

```
ava.io.*;

import java.util.Scanner;

public class Main {

    static int roomNum; // declare variables for methods.
    static String roomName;

    public static void main(String[] args) {

        String[] hotel = new String[8]; // creating arrays
        String[] firstName = new String[8];
        String[] surName = new String[8];
        int[] creditNum = new int[8];
        int[] guestNum = new int[8];

        initialise(hotel,firstName,surName,creditNum,guestNum); //initialise
        methods

        while (true) {
            System.out.println("-----Menu-----");
            System.out.println("A: Add customer rooms.");
            System.out.println("V: View all Rooms.");
```

```
System.out.println("E: Display Empty Rooms.");
System.out.println("D: Delete customer from room.");
System.out.println("F: Find room from customer name.");
System.out.println("S: Store program data in to file.");
System.out.println("L: Load program data from file.");
System.out.println("Q: View guests Ordered alphabetically by
name.");
System.out.println("-----");
```

```
System.out.println();
Scanner scn = new Scanner(System.in);
System.out.println("Enter the suggestion from menu : ");
String menu = scn.nextLine();
menu = menu.toUpperCase(); // changing letters to uppercase

switch (menu) {
    case "A":
        add(hotel,firstName,surName,creditNum,guestNum); //using
switch for menu

        break;
    case "V":
        view(hotel,firstName,surName,creditNum,guestNum);

        break;
    case "E":
        emptyroom(hotel);
```



```

        break;
    case "D":

delete_customer(hotel,firstName,surName,creditNum,guestNum);

        break;
    case "F":
        find_customer(hotel);

        break;
    case "S":
        storedata(hotel,firstName,surName,creditNum,guestNum);

        break;
    case "L":
        loaddata();

        break;
    case "Q":
        guests_alpha(hotel);

        break;
    default:
        System.out.println("Invalid Suggestion");

    }
}

```

```
}
```

```
private static void initialise(String[] surName, String[] hotel, String[]  
firstName, int[] creditNum, int[] guestNum){ // getting the empty array
```

```
for (int i = 0; i < 8; i++ ){
```

```
    hotel[i] = "empty";
```

```
    firstName[i]="empty";
```

```
    surName[i]="empty";
```

```
    creditNum[i]=0;
```

```
    guestNum[i]=0;
```

```
}
```

```
}
```

```
private static void add(String[] hotel, String[] firstName, String[] surName,  
int[] creditNum, int[] guestNum) { // adding customer rooms
```

```
    Scanner scn = new Scanner(System.in);
```

```
    System.out.println("Enter room number (0-7) or 8 to stop:");
```

```
    roomNum = scn.nextInt();
```

```
    System.out.println("Enter name for room " + roomNum + " :");
```

```
    String roomName = scn.next();
```

```
    hotel[roomNum]=roomName;
```

```
    System.out.println("Enter the first name: ");
```

```
    String fName = scn.next();
```

```
firstName[roomNum]=fName;
```

```
System.out.println("Enter the surname: ");
```

```
String sName = scn.next();
```

```
surName[roomNum]=sName;
```

```
System.out.println("Enter the credit card number: ");
```

```
int cNum=scn.nextInt();
```

```
creditNum[roomNum]=cNum;
```

```
System.out.println("Enter the number of guest: ");
```

```
int gNum=scn.nextInt();
```

```
guestNum[roomNum]=gNum;
```

```
}
```

```
private static void view(String[] hotel, String[] firstName, String[]  
surName, int[] creditNum, int[] guestNum) {    // viewing the data
```

```
    for (int i = 0; i < 8; i++) {
```

```
        System.out.println("room " + i + " occupied by " + hotel[i]);
```

```
        System.out.println(firstName[i]);
```

```
        System.out.println(surName[i]);
```

```
        System.out.println(creditNum[i]);
```

```
        System.out.println(guestNum[i]);
```

```
    }
```

```
}
```

```

private static void emptyroom(String[] hotel) {    //getting the empty array
    for (int i = 0; i < 8; i++ )
    {
        if (hotel[i] == "empty") {
            System.out.println("room " + i + " is empty");
        }
    }
}

```

```

private static void delete_customer(String[] hotel, String[] firstName,
String[] surName, int[] creditNum, int[] guestNum) {    // deleting the data from
the array

```

```

    Scanner scn = new Scanner(System.in);
    System.out.println("Enter room number to delete(1-8):");
    roomNum = scn.nextInt() ;
    hotel[roomNum]= "empty";
    firstName[roomNum]="empty";
    surName[roomNum]="empty";
    creditNum[roomNum]=0;
    guestNum[roomNum]=0;

    System.out.println("Room Deleted");
}

```

```

private static void find_customer(String[] hotel) {    // finding the matching
customer name

```

```

    Scanner scn= new Scanner(System.in);

```

```

System.out.println("Enter the name: ");
roomName= scn.nextLine();
boolean finding= false;
for (int i=0; i<8;i++) {
    if (roomName.equals(hotel[i])) {
        System.out.println("Name matched with the customer");
        finding = true;
    }
}
if(finding==false) {
    System.out.println("there's no room book with that name");
}
}

```

```

private static void storedata(String[] hotel, String[] firstName, String[]
surName, int[] creditNum, int[] guestNum) { //storing data

```

```

    try (PrintWriter get = new PrintWriter(new
FileWriter("E:\\CW\\sd2\\task_3(array)\\store.txt"))) { //path for store data

```

```

        for (int i = 0; i < 8; i++) {
            get.println("Name and Room number is: " + hotel[i] + " at: " + i);
            get.println("First name is: " + firstName[i]);
            get.println(" Surname is: " + surName[i] );
            get.println(" Credit card number is: " + creditNum[i]);
            get.println("Number of guests is: " + guestNum[i]);
        }
    } catch (IOException e) {
        e.printStackTrace();
    }
}

```

```

    }

    System.out.println("All data updated.");

}

private static void loaddata() {    //loading data
    try {
        File myObj = new File("E:\\CW\\sd2\\task_3(array)\\store.txt");
//path for load data
        Scanner scn = new Scanner(myObj);
        while (scn.hasNextLine()) {
            String load = scn.nextLine();
            System.out.println(load);
        }
        scn.close();
    }
    catch (FileNotFoundException e) {
        System.out.println(" Error .");
        e.printStackTrace();
    }
}

private static void guests_alpha(String[] hotel) {    //finding the alphabetical
oder of names

    String[] guest = new String[hotel.length];
    for (int a=0;a< hotel.length;a++){
        guest[a]=hotel[a];
    }
}

```

```

for (int i = 0; i < 8; i++) {

    for (int j = i + 1; j < 8; j++) { // using two for loops for checking

        if (guest[i].compareTo(guest[j]) > 0) {

            String get = guest[i];
            guest[i] = guest[j];
            guest[j] = get;
        }
    }
}

System.out.println("guests names in alphabetical order: ");
for (int i = 0; i < 8; i++) {
    System.out.println(hotel[i]);
}

}

```

2.Class version

2.1 Hotel class

```
import java.io.*;
import java.util.Scanner;

public class Hotel {
    static int roomNum;    // declare variables for methods.
    static String roomName;
    public static void main(String[] args) {
        Room[] hotel = new Room[8]; // creating the array
        hotel[0] = new Room();    //adding elements to array
        hotel[1] = new Room();
        hotel[2] = new Room();
        hotel[3] = new Room();
        hotel[4] = new Room();
        hotel[5] = new Room();
        hotel[6] = new Room();
        hotel[7] = new Room();
        initialise(hotel);

        while (true) {
            System.out.println("-----Menu-----");
            System.out.println("A: Add customer rooms.");
            System.out.println("V: View all Rooms.");
```



```
System.out.println("E: Display Empty Rooms.");
System.out.println("D: Delete customer from room.");
System.out.println("F: Find room from customer name.");
System.out.println("S: Store program data in to file.");
System.out.println("L: Load program data from file.");
System.out.println("Q: View guests Ordered alphabetically by name.");
System.out.println("-----");
```

```
System.out.println();
Scanner scn = new Scanner(System.in);
System.out.println("Enter the suggestion from menu : ");
String menu = scn.nextLine();
menu = menu.toUpperCase(); // changing letters to uppercase
```

```
switch (menu) { // using switch for menu
    case "A":
        add(hotel); // adding customer rooms

        break;
    case "V":
        view(hotel);

        break;
    case "E":
        emptyroom(hotel);
```

```
        break;
    case "D":
        delete_customer(hotel);

        break;
    case "F":
        find_customer(hotel);

        break;
    case "S":
        storedata(hotel);

        break;
    case "L":
        loaddata();

        break;
    case "Q":
        guests_alpha(hotel);

        break;
    default:
        System.out.println("Invalid Suggestion");

    }
}
}
```

```
private static void initialise(Room[] hotelRef){ // getting the empty array

    for (int i = 0; i < 8; i++ ){
        hotelRef[i].setName("empty");
        hotelRef[i].setFirst_name("empty");
        hotelRef[i].setSurName("empty");
        hotelRef[i].setCredit_num(0);
        hotelRef[i].setNum_of_guests(0);

    }
}
```

```
private static void add(Room[] hotelref) { // adding customer rooms

    Scanner scn = new Scanner(System.in);
    System.out.println("Enter room number (0-7) or 8 to stop:");
    roomNum = scn.nextInt();

    System.out.println("Enter name for room " + roomNum + " :");
    String roomName = scn.next();
    hotelref[roomNum].setName(roomName);

    System.out.println("Enter the first name: ");
    String fName = scn.next();
    hotelref[roomNum].setFirst_name(fName);

    System.out.println("Enter the surname: ");
```

```

String sName = scn.next();
hotelref[roomNum].setSurName(sName);

System.out.println("Enter the credit card number: ");
int cNum=scn.nextInt();
hotelref[roomNum].setCredit_num(cNum);

System.out.println("Enter the number of guest: ");
int gNum=scn.nextInt();
hotelref[roomNum].setNum_of_guests(gNum);

}

private static void view(Room[] hotelref) {    // viewing the data
    for (int i = 0; i < 8; i++) {
        System.out.println("room " + i + " occupied by " +
hotelref[i].getName());
        System.out.println(hotelref[i].getFirst_name());
        System.out.println(hotelref[i].getSurName());
        System.out.println(hotelref[i].getCredit_num());
        System.out.println(hotelref[i].getNum_of_guests());
    }
}

private static void emptyroom(Room[] hotelref) { //getting the empty array
    for (int i = 0; i < 8; i++ )
    {

```

```

        if (hotelref[i].getName().equals("empty")){
            System.out.println("room " + i + " is empty");}
    }

}

```

private static void delete_customer(Room[] hotelref) { // deleting the data from the array

```

    Scanner scn = new Scanner(System.in);
    System.out.println("Enter room number to delete(1-8):");
    roomNum = scn.nextInt() ;
    hotelref[roomNum].setName("empty");
    hotelref[roomNum].setFirst_name("empty");
    hotelref[roomNum].setSurName("empty");
    hotelref[roomNum].setCredit_num(0);
    hotelref[roomNum].setNum_of_guests(0);
    System.out.println("Room Deleted");
}

```

private static void find_customer(Room[] hotelref) { // finding the matching customer name

```

    Scanner scn= new Scanner(System.in);
    System.out.println("Enter the name: ");
    roomName= scn.nextLine();
    boolean finding= false;
    for (int i=0; i<8;i++) {
        if (roomName.equals(hotelref[i].getName())) {
            System.out.println("Name matched with the customer");

```

```

        finding = true;
    }
}
if(finding==false) {
    System.out.println("there's no room book with that name");
}
}

private static void storedata(Room[] hotelref) {    //storing data
    try (PrintWriter get = new PrintWriter(new
    FileWriter("E:\\CW\\sd2\\task_3\\store.txt"))) { //path for store

        for (int i = 0; i < 8; i++) {

            get.println("Name and Room number is: " + hotelref[i].getName() + "
at: " + i);

            get.println("First name is: " + hotelref[i].getFirst_name());
            get.println("Surname is: " + hotelref[i].getSurName());
            get.println("Credit card number is: " + hotelref[i].getCredit_num());
            get.println("Number of guests is: " + hotelref[i].getNum_of_guests());

        }
    } catch (IOException e) {
        e.printStackTrace();
    }
    System.out.println("All data updated. ");

}

private static void loaddata() { //loading data

```

```

try {
    File myObj = new File("E:\\CW\\sd2\\task_3\\store.txt"); //path for
load data
    Scanner scn = new Scanner(myObj);
    while (scn.hasNextLine()) {
        String load = scn.nextLine();
        System.out.println(load);
    }
    scn.close();
}
catch (FileNotFoundException e) {
    System.out.println(" Error .");
    e.printStackTrace();
}
}

```

private static void guests_alpha(Room[] hotelRef) { //finding the alphabetical oder of names

```

Room[] guest = new Room[8];
for (int a=0;a< hotelRef.length;a++){
    guest[a]=hotelRef[a];
}
for (int i = 0; i <8; i++) {    // using two for loops for checking

    for (int j = i + 1; j <8; j++) {

        if (guest[i].getName().compareTo(guest[j].getName()) > 0) {

```

```

        String get = guest[i].getName();
        guest[i].setName(guest[j].getName());
        guest[j].setName(get);
    }
}

```

```

System.out.println("guests names in alphabetical order: ");
for (int i = 0; i <8; i++) {
    System.out.println(hotelRef[i].getName());
}
}

```

2.2 Room

```

public class Room extends Person { //connect room and person rooms

```

```

    public String cusName; // declare variables for room class

```

```

    public void setName(String samName) { //

```

```

        cusName=samName;
    }

```

```

    public String getName() {

```



```
        return cusName;
    }
}
```

2.3 Person class

```
public class Person {
    private String first_name;    //declare variables for person room get sets
    private int num_of_guests;
    private String surName;
    private int credit_num;

    public String getFirst_name() {
        return first_name;
    }    //get for first name

    public void setFirst_name(String first_name) {    // set for first name
        this.first_name = first_name;
    }

    public int getNum_of_guests() { return num_of_guests; }    //get for number
of guests

    public void setNum_of_guests(int num_of_guests) { this.num_of_guests =
num_of_guests; }    // set for number of guests
}
```

```
public String getSurName() {  
    return surName;  
} //get for surname  
  
public void setSurName(String surName) {  
    this.surName = surName;  
} //set for surname  
  
public int getCredit_num() {  
    return credit_num;  
} //get for credit card number  
  
public void setCredit_num(int credit_num) {  
    this.credit_num = credit_num;  
} //set for credit card number  
}
```

3.Screen shots

Add customer for rooms.

```
-----Menu-----
A: Add customer rooms.
V: View all Rooms.
E: Display Empty Rooms.
D: Delete customer from room.
F: Find room from customer name.
S: Store program data in to file.
L: Load program data from file.
Q: View guests Ordered alphabetically by name.
-----

Enter the suggestion from menu :
a
Enter room number (0-7) or 8 to stop:
0
Enter name for room 0 :
kaveesh
Enter the first name:
kaveesh
Enter the surname:
fernando
Enter the credit card number:
1234
Enter the number of guest:
2
```

```
-----Menu-----  
A: Add customer rooms.  
V: View all Rooms.  
E: Display Empty Rooms.  
D: Delete customer from room.  
F: Find room from customer name.  
S: Store program data in to file.  
L: Load program data from file.  
Q: View guests Ordered alphabetically by name.  
-----
```

Enter the suggestion from menu :

a

Enter room number (0-7) or 8 to stop:

1

Enter name for room 1 :

anupa

Enter the first name:

anupa

Enter the surname:

rathnayaka

Enter the credit card number:

4567

Enter the number of guest:

3

```
-----Menu-----  
A: Add customer rooms.  
V: View all Rooms.  
E: Display Empty Rooms.  
D: Delete customer from room.  
F: Find room from customer name.  
S: Store program data in to file.  
L: Load program data from file.  
Q: View guests Ordered alphabetically by name.  
-----
```

Enter the suggestion from menu :

a

Enter room number (0-7) or 8 to stop:

2

Enter name for room 2 :

akila

Enter the first name:

akila

Enter the surname:

srikantha

Enter the credit card number:

5678

Enter the number of guest:

4

View all rooms

Enter the suggestion from menu :

✓

room 0 occupied by kaveesh

kaveesh

fernando

1234

2

room 1 occupied by anupa

anupa

rathnayaka

4567

3

room 2 occupied by akila

akila

srikantha

5678

4

room 3 occupied by empty

empty

empty

0

0

room 4 occupied by empty

empty

empty

0

0

room 5 occupied by empty

empty

empty

0

0

room 6 occupied by empty

empty

empty

0

0

room 7 occupied by empty

empty

empty

Display empty rooms

```
-----Menu-----  
A: Add customer rooms.  
V: View all Rooms.  
E: Display Empty Rooms.  
D: Delete customer from room.  
F: Find room from customer name.  
S: Store program data in to file.  
L: Load program data from file.  
Q: View guests Ordered alphabetically by name.  
-----
```

Enter the suggestion from menu :

e

```
room 3 is empty  
room 4 is empty  
room 5 is empty  
room 6 is empty  
room 7 is empty
```

Delete customer from room

```
-----Menu-----  
A: Add customer rooms.  
V: View all Rooms.  
E: Display Empty Rooms.  
D: Delete customer from room.  
F: Find room from customer name.  
S: Store program data in to file.  
L: Load program data from file.  
Q: View guests Ordered alphabetically by name.  
-----
```

Enter the suggestion from menu :

d

Enter room number to delete(1-8):

1

Room Deleted

Enter the suggestion from menu :

✓

room 0 occupied by kaveesh

kaveesh

fernando

1234

2

room 1 occupied by empty

empty

empty

0

0

room 2 occupied by akila

akila

srikantha

5678

4

room 3 occupied by empty

empty

empty

0

0

room 4 occupied by empty

empty

empty

0

0

room 5 occupied by empty

empty

empty

0

Find room from customer name

```
-----Menu-----  
A: Add customer rooms.  
V: View all Rooms.  
E: Display Empty Rooms.  
D: Delete customer from room.  
F: Find room from customer name.  
S: Store program data in to file.  
L: Load program data from file.  
Q: View guests Ordered alphabetically by name.  
-----
```

Enter the suggestion from menu :

f

Enter the name:

kaveesh

Name matched with the customer

Store program data

Enter the suggestion from menu :

s

All data updated.

-----Menu-----

A: Add customer rooms.

V: View all Rooms.

E: Display Empty Rooms.

D: Delete customer from room.

F: Find room from customer name.

S: Store program data in to file.

L: Load program data from file.

Q: View guests Ordered alphabetically by name.

Load program data

```
Enter the suggestion from menu :  
[  
Name and Room number is: kaveesh at: 0  
First name is: kaveesh  
Surname is: fernando  
Credit card number is: 1234  
Number of guests is: 2  
Name and Room number is: empty at: 1  
First name is: empty  
Surname is: empty  
Credit card number is: 0  
Number of guests is: 0  
Name and Room number is: akila at: 2  
First name is: akila  
Surname is: srikantha  
Credit card number is: 5678  
Number of guests is: 4  
Name and Room number is: empty at: 3  
First name is: empty  
Surname is: empty  
Credit card number is: 0  
Number of guests is: 0
```

View guest ordered alphabetically by name

```
-----Menu-----  
A: Add customer rooms.  
V: View all Rooms.  
E: Display Empty Rooms.  
D: Delete customer from room.  
F: Find room from customer name.  
S: Store program data in to file.  
L: Load program data from file.  
Q: View guests Ordered alphabetically by name.  
-----
```

Enter the suggestion from menu :

```
q  
guests names in alphabetical order:  
kaveesh  
empty  
akila  
empty  
empty  
empty  
empty  
empty
```

REFERENCES

W3schools.com. (2019). Java Classes and Objects. [online] Available at: https://www.w3schools.com/java/java_classes.asp. [accessed 30st march 2021].

W3schools.com. (2019). Java Arrays. [online] Available at: https://www.w3schools.com/java/java_arrays.asp. [accessed 25 rd march 2021].

W3schools.com. (2020). Java Create and Write To Files. [online] Available at: https://www.w3schools.com/java/java_files_create.asp. [accessed 1 april 2021].

Stack Overflow. (n.d.). java - How to sort a String array alphabetically (without using compareTo or Arrays.sort). [online] Available at: <https://stackoverflow.com/questions/18689672/how-to-sort-a-string-array-alphabetically-without-using-compareto-or-arrays-sor/52347693> [Accessed 29 march 2021].