Ezzeldin el Galab
900211996

Assignment 2 Report

Created an abstract Person class with a pure virtual printInfo function that is overrided in all its derived classes as well having an Appointment struct composed of hours and minutes. Followed are Customer and Mechanic classes derived from Person with each having their own setter, getters and virtual printInfo function. Customer has overloaded operators to compare between customer appointments and Mechanic has an isAvailable function that validates that the mechanic is free at a certain appointment. Finally, a template queue class that works similarly to linked lists using nodes is made with several functions including pop, push, Front, isFull, and getSize.

In the main the program starts by asking the user whether he would like to enter the values for each customer and mechanic manually, have them generated automatically, or be read from a file. The first two work exactly the same other than the way the input is made. Program asks the user for the number of mechanics and customers to be made and 2 dynamic arrays are respectively made which are then filled manually(or automatically) by the user. Program then uses a nested for loop and the mechanic isAvailable function to assign each customer a mechanic at his set appointment and if there are no free mechanics at the desired appointment then we send a message informing them that the appointment is cancelled. The customer array is then sorted(bubble sort) from earliest to latest appointment using overloaded operators and pushed in order to a created queue. After each run of the program the customers and mechanics are stored into their respective text files and can be read through the third method. The third method which is reading from files checks whether the file is empty first and if so prompts the user to either run the any of the other two methods first or to put some information into the text files manually. Then it goes through the file line by line. First line indicates the number of mechanics and customers in the file for the program to loop through them. Then it takes in order the name, id, age, and appointments of each and fills an array of mechanics and customers set to these values. Similar to the other two method it sorts them into the queue then prints the information. At the end of each run of the program the queue is deconstructed, and the arrays are deleted to free up memory.

Method1(Manually):

Ezzeldin el Galab
900211996

```
Enter 1 to manually enter the information, 2 to generate it randomly, 3 to read it from a file: 1
Enter the number of customers: 5
Enter the number of mechanics: 3
Enter Customer 1's Name: Ezz
Enter Customer 1's ID: 1
Enter Customer 1's Age: 1
Enter Customer 1's Appointment:
1. 9:30
2. 10:30
3. 11:30
1
Enter Customer 2's Name: Omar
Enter Customer 2's ID: 2
Enter Customer 2's Age: 2
Enter Customer 2's Appointment:
1. 9:30
2. 10:30
3. 11:30
3
Enter Customer 3's Name: Mohamed
Enter Customer 3's ID: 3
Enter Customer 3's Age: 3
Enter Customer 3's Appointment:
1. 9:30
2. 10:30
3. 11:30
3
Enter Customer 4's Name: Amar
Enter Customer 4's ID: 4
Enter Customer 4's Age: 4
```

```
Enter Customer 5's ID: 5
Enter Customer 5's Age: 5
Enter Customer 5's Appointment:
1. 9:30
2. 10:30
3. 11:30
3
Enter Mechanic 1's Name: Ahmed
Enter Mechanic 1's ID: 1
Enter Mechanic 1's Age: 1
Enter Mechanic 2's Name: Ziad
Enter Mechanic 2's ID: 2
Enter Mechanic 2's Age: 2
Enter Mechanic 3's Name: Salama
Enter Mechanic 3's ID: 3
Enter Mechanic 3's Age: 3
Ezz has an appointment at 9:30 with Ahmed

Amar has an appointment at 9:30 with Ziad

Omar has an appointment at 11:30 with Ahmed

Mohamed has an appointment at 11:30 with Ziad

Mona has an appointment at 11:30 with Salama


C:\Users\M.H\Desktop\University\Assignments\Assignment 2\x64\Debug\Assignment 2.exe (process 23300) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
```

Method2(Automatically):

Ezzeldin el Galab
900211996

```
Enter 1 to manually enter the information, 2 to generate it randomly, 3 to read it from a file: 2
Enter the number of customers: 10
Enter the number of mechanics: 3
Customer 1 has an appointment at 9:30 with Mechanic 1

Customer 9 has an appointment at 9:30 with Mechanic 2

Customer 2 has an appointment at 10:30 with Mechanic 1

Customer 3 has an appointment at 10:30 with Mechanic 2

Customer 4 has an appointment at 11:30 with Mechanic 1

Customer 5 has an appointment at 11:30 with Mechanic 2

Customer 6 has an appointment at 11:30 with Mechanic 3

No mechanic available for customer Customer 7 at 11:30

No mechanic available for customer Customer 8 at 11:30

No mechanic available for customer Customer 10 at 11:30


C:\Users\M.H\Desktop\University\Assignments\Assignment 2\x64\Debug\Assignment 2.exe (process 9300) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .
```

Method3(Reading from files):

```
Enter 1 to manually enter the information, 2 to generate it randomly, 3 to read it from a file: 3
Customer 1 has an appointment at 10:30 with Mechanic 1

Customer 3 has an appointment at 10:30 with Mechanic 2

Customer 4 has an appointment at 10:30 with Mechanic 3

No mechanic available for customer Customer 5 at 10:30

Customer 2 has an appointment at 11:30 with Mechanic 1


C:\Users\M.H\Desktop\University\Assignments\Assignment 2\x64\Debug\Assignment 2.exe (process 19284) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .
```

This is using the information generated from a method 2 but the text files can also be filled manually through method 1 or directly editing the .txt

Format that the programs reads the information in:
customers.txt {

//number of customers

//name

Ezzeldin el Galab
900211996

//id

//appointment.hours

//appointment.min

}

Mechanics.txt {

//number of mechanics

//name

//id

//age

//number of appointments

//appointment.hours

//appointment.mins

}