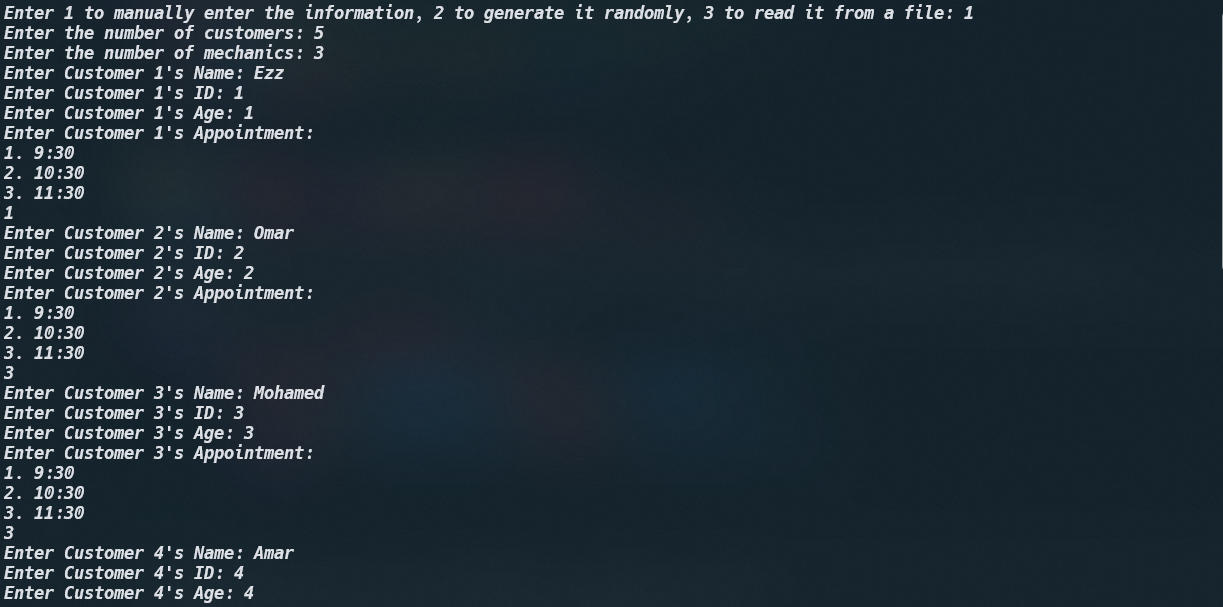
Assignment 2 Report

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.

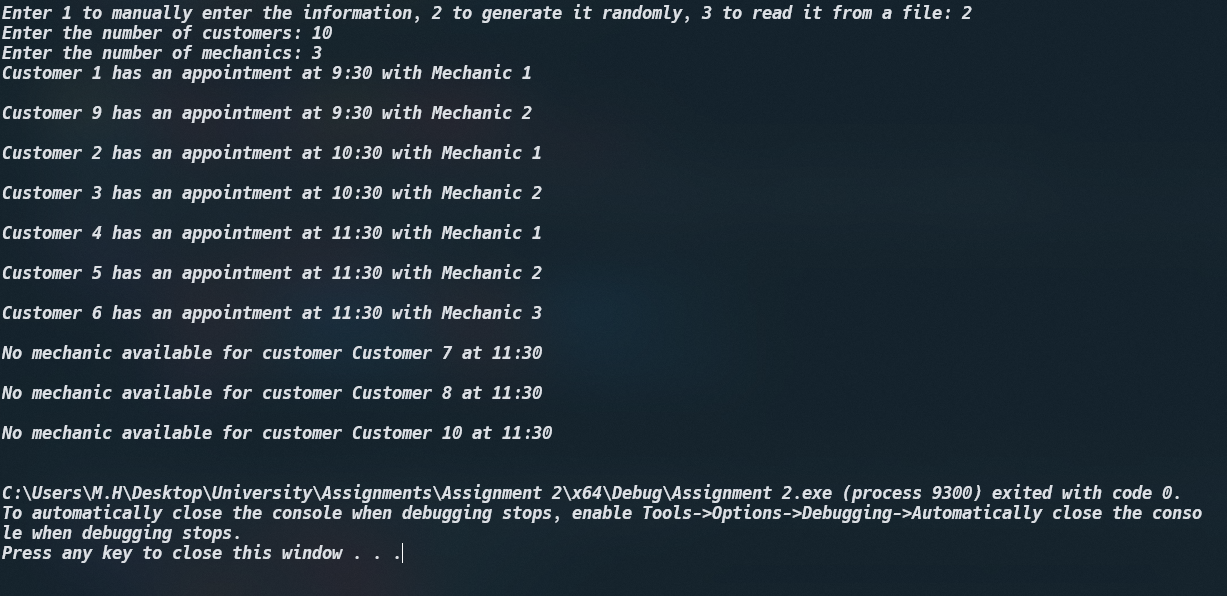
Method1(Manually):



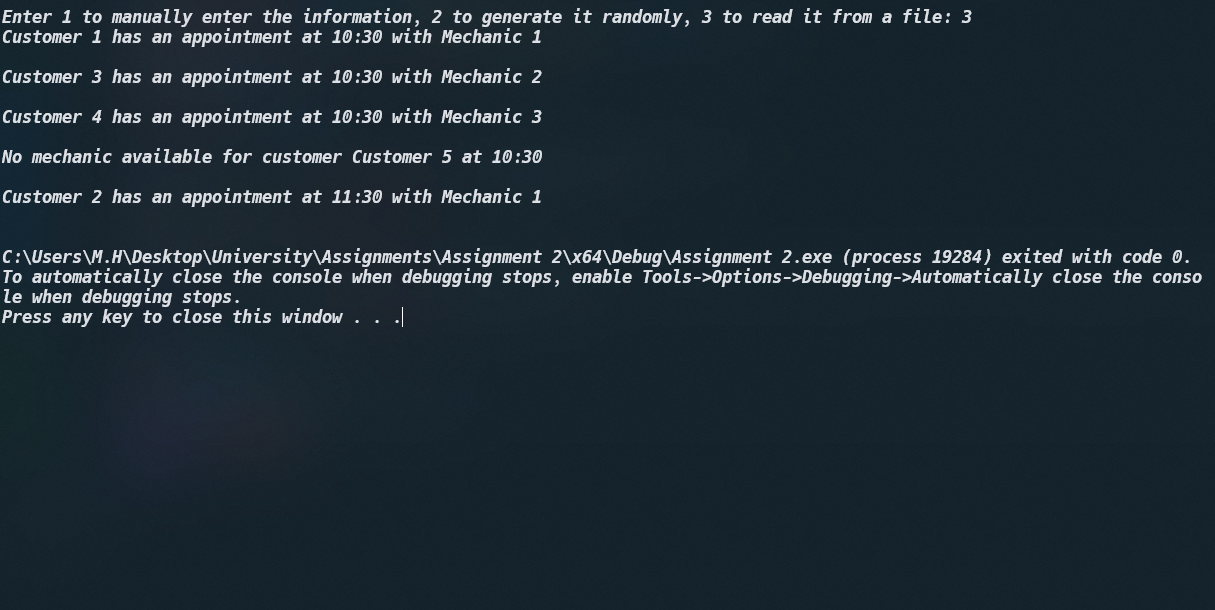
Graphical user interface, text, website

Description automatically generated

Method2(Automatically):



Method3(Reading from files):



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

//appointment.hours

//appointment.min

}

Mechanics.txt {

//number of mechanics

//name

//id

//age

//number of appointments

//appointment.hours

//appointment.mins

}