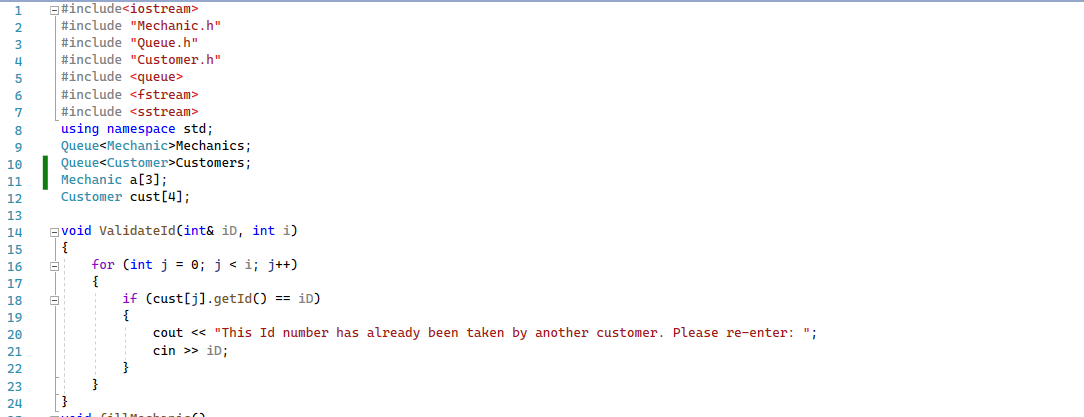
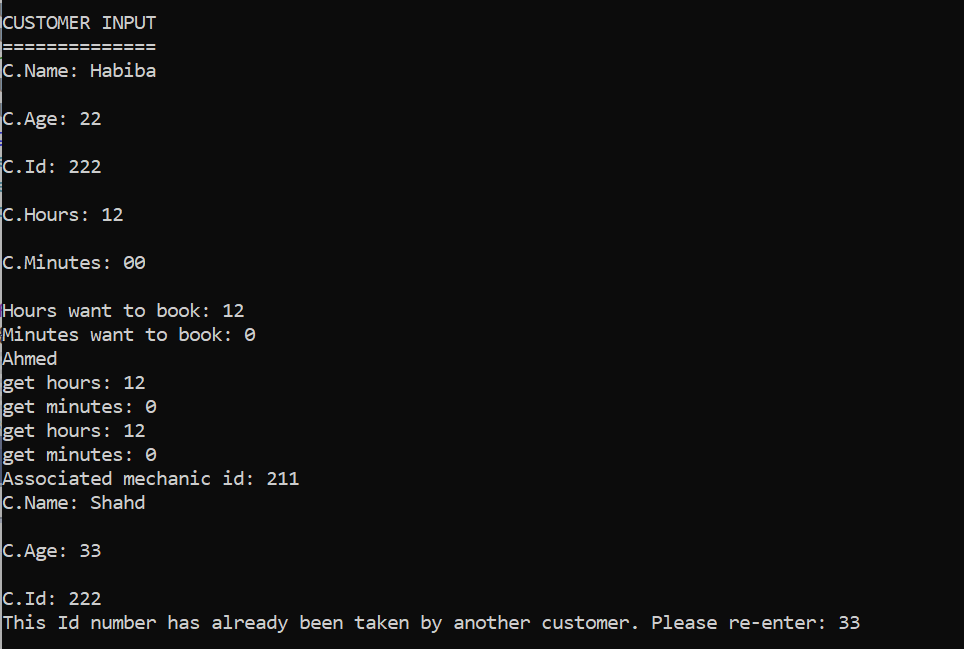
REPORT ASSIGNMENT 1

1. This is the validation of the customer Id. Where one customer cannot have the same ID of a previous customer.

* In code:

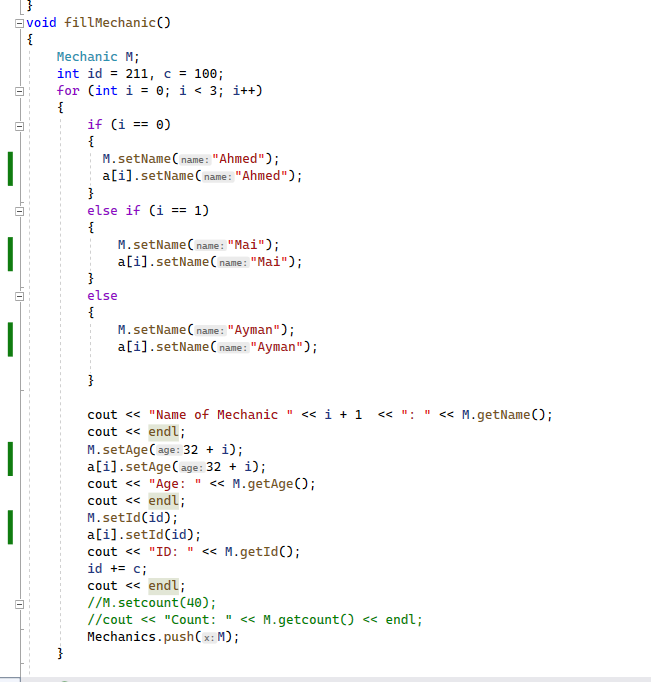


* In terminal:

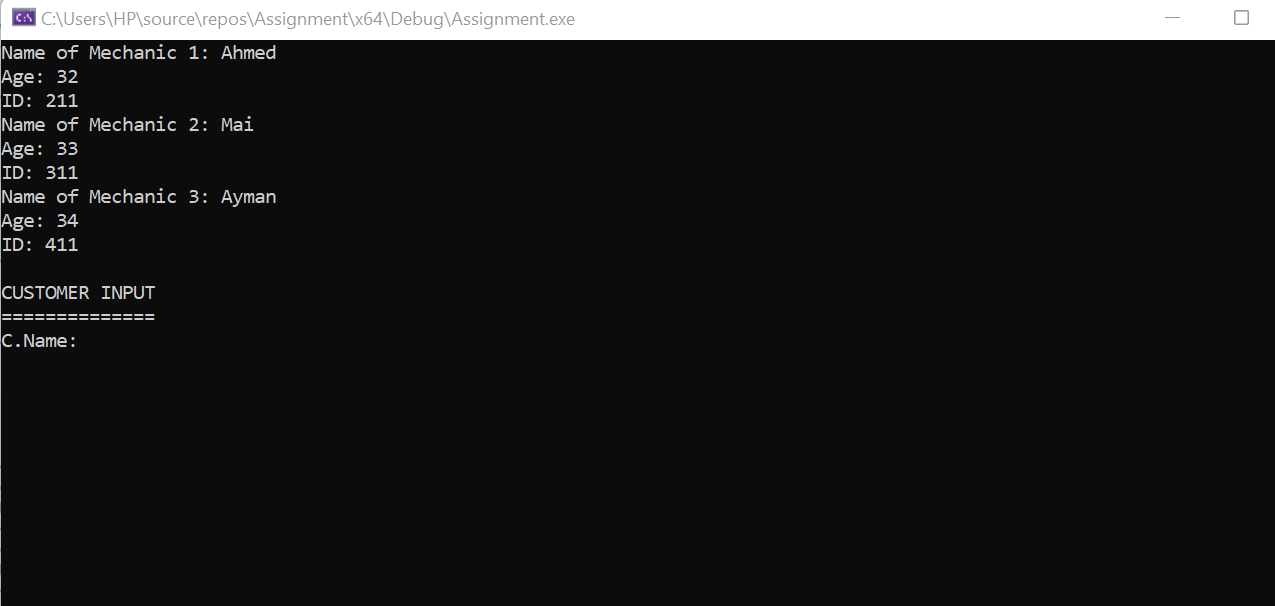


1. The filling of the mechanic queue:

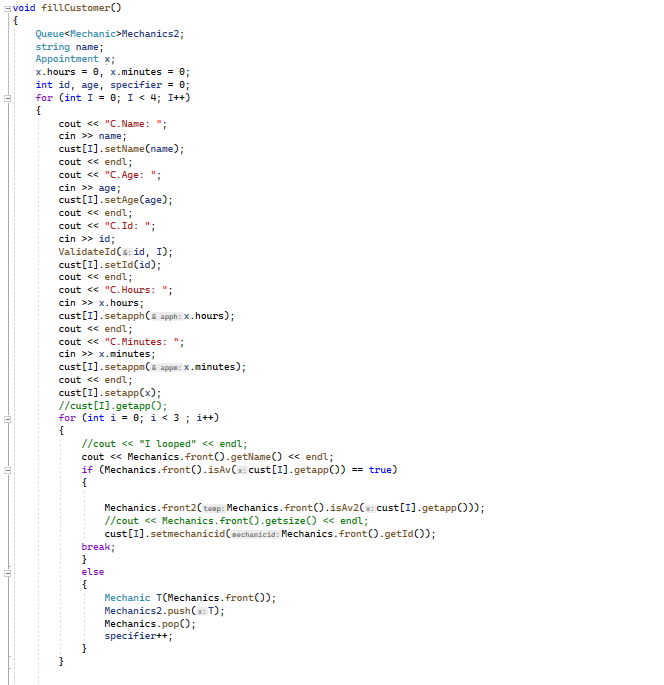
:The code



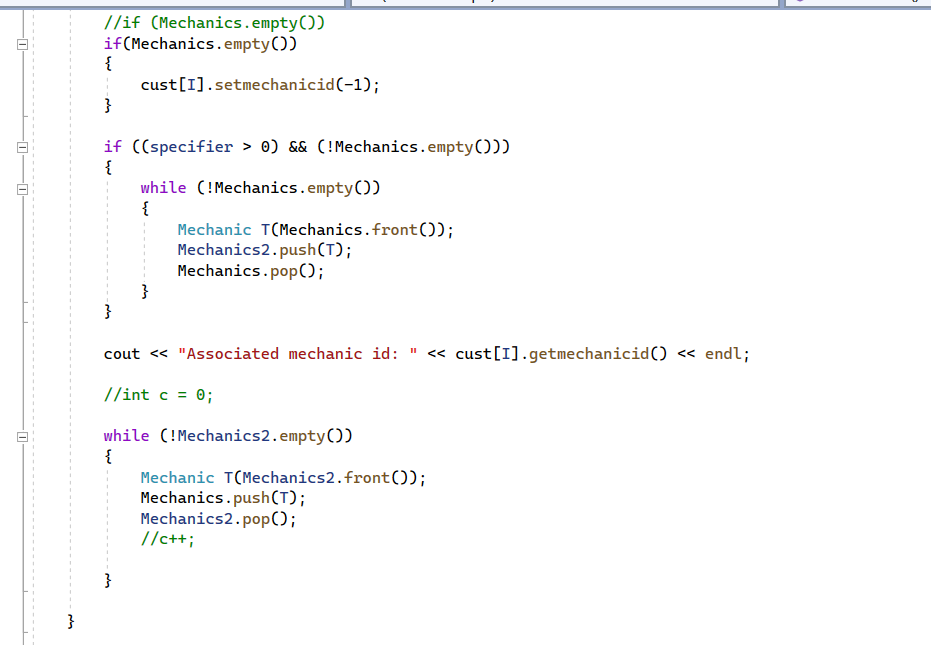
Here we used one temporary Mechanic object that gets looped over 3 times in order to enter the descriptions of the 3 mechanics and cout them in the terminal. Before we move to couting and filling the information of the next customer, we push the temp object to the Mechanics queue.

* In the terminal

1. The filling of the customer



Here we will fill the customer array. We will enter the data of 4 customers, where it has name, age, Id, hours and minute that they want to register. After the 4 elements were entered in the customer, we check if the first mechanic is available or not. This will only return a bool function which only checks true or false. If true, then we will pass Mechanics front to Av2 which pushes back the appointment and returns (Mechanics front) after it got modified with the added appointment. Then we will set the Id of the mechanic front to the set mechanic id.

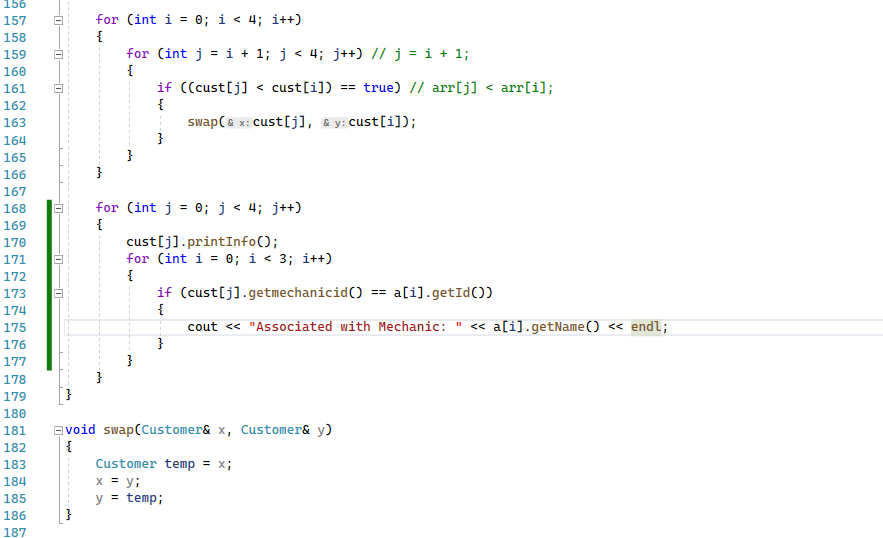
If it is false we will pop and temporarily store whatever that was popped in a temporary Mechanic queue and then reloop 3 times. If isAv is true therefore we break. But since it entered the else, then the specifier got incremented which means the following:

If the queue is empty then this means that there is no Mechanic available. Therefore we set mechanic id to -1 since there are no reservation.

If the specifier is bigger than 0 and then mechanic queue is not empty then this means that a mechanic got chosen but not the first one. We will empty the leftover of Mechanics queue and add it to Mechanics2 queue.

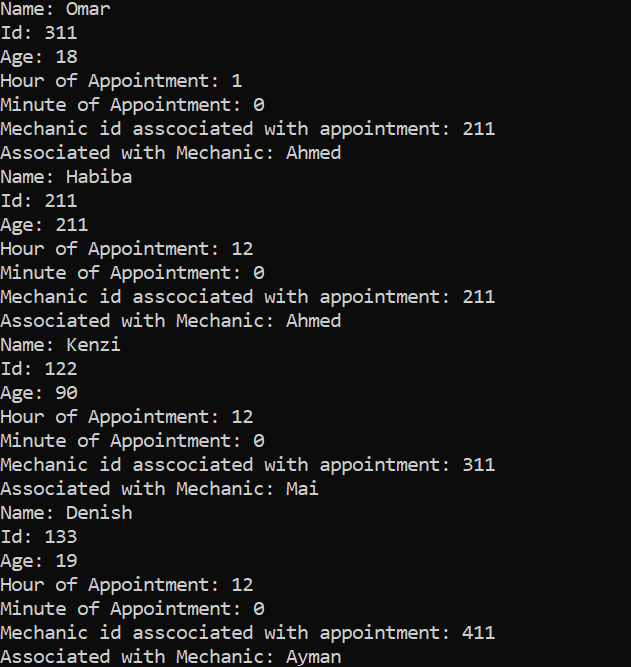
Then we move to the last while loop. Where we move whatever is in Mechanics2 queue back to Mechanics queue.

1. Organising the appointments ascending order:



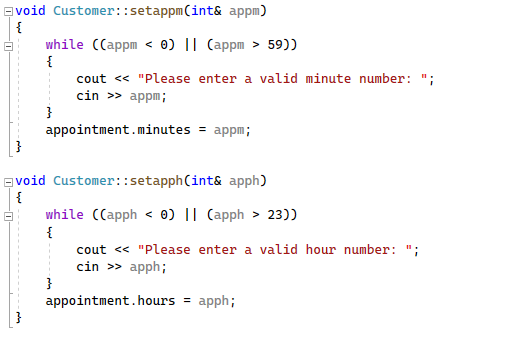
Here we are organising the appointments that each customer reserved and printing out its details with their associated mechanic id. (we organise them ascendingly) – from earliest to latest. And here we use the array to easily spot out the name of the mechanic with the same id association

Therefore, this will be in the terminal:

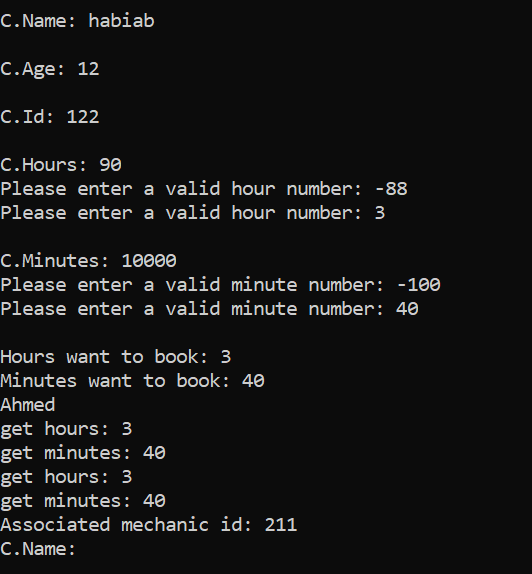


1. Hours and minutes validation

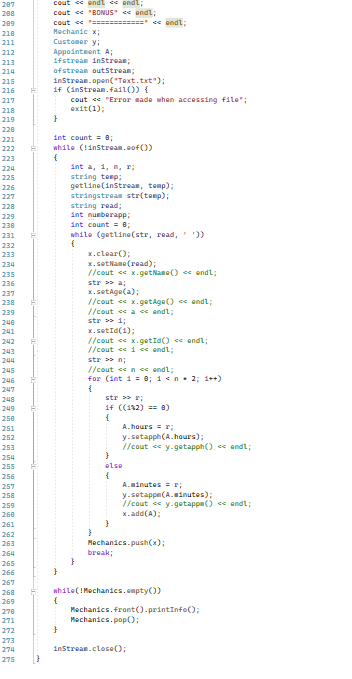
Code:



Terminal:



Bonus is in the rest of the main. Used parsing in order to read the input from the text.

the line whilegetline(str, read, ‘ ‘)

is a three parameter function that takes an extra parameter which indicates the delimiter that the parsing will be based upon.

We take the first word as the name. Therefore not changing string into an integer.

Next we should change the consecutive strings into integers since they represent the id, age and the number of appointments. Each will be stored in a different int variable.

We place the string (the string from the text) to the int variable (also internally changing it) by doing so

Str >> variable name;

Then there will be a for loop of n multiplied by 2. If the count modulus 2 is equal to zero than the str represents an hour

Then we push the temporary Mechanic

If not then it represents a minute.

Will do this with everyline until the file is empty.

Then we print its info and pop the queue.