**PROBLEM STATEMENT**

This program will calculate the total time that a vehicle was parked and how much it cost to park in that garage given the type of vehicle and the time in and time out of the garage.

**Output Display**

Car Bus Truck

First 3 hours First 2 hours First hour  
0.00 2.00 3.75

After 3 hours After 2 hours After first hour  
1.25 per hour 2.50 per hour 4.50 per hour

NO OVERNIGHT PARKING  
--------------------------------------------------------------------------------------

Type of vehicle:

Hours in:

Minutes in:

Hours out:

Minutes out:  
--------------------------------------------------------------------------------------

Type of vehicle:

Time in:

Time out:  
--------------------------------------------------------------------------------------

Total time parked:

Total Charges:

**Structure Chart**

Main

PrintReport

Charge

Rate

Time

GetUserInfo

Flowchart for Main

Start

GetUserInfo

Time

Rate

Charge

printReport

End

Pseudo-Code for Main

Start of algorithm for Main

1. GetUserInfo();
2. Time();
3. Rate();
4. Charge();
5. PrintReport();

End of algorithm for Main

Flowchart for getUserInfo

End

While (minuteOut < 0 || minuteOut > 59) invalid

Prompt user and ask for minute out

While (hourOut < 0 || hourOut > 23) invalid

While (minuteIn < 0 || minuteIn > 59) invalid

Prompt user and ask for hour out

Prompt user and ask for minute in

While (hourIn < 0 || hoursIn > 23)   
invalid

Prompt user and ask for hour in

While (vehicle != ‘c’ || ‘b’ || ‘t’)   
invalid

Prompt user and ask for type of vehicle

Start

Pseudo Code for getUserInfo

Start of algorithm for Get userInfo

1. Prompt user and get ask for vehicle  
- while (vehicle != ‘c’ || !=’C’ || != ‘b’ || != ‘B’ || != ‘t’ || != ‘T’)  
{  
 cout << “Invalid input. Please try again.”;  
 cin >> vehicle;  
}  
2. Prompt user and get hourIn  
- while (hourIn < 0 || hourIn > 23)  
{  
 cout << “Invalid input. Please try again.”;  
 cin >> hourIn;  
}  
3. Prompt user and get minuteIn  
- while (minuteIn < 0 || minuteIn > 59)  
{  
 cout << “Invalid input. Please try again.”;  
 cin >> minuteIn;  
}  
4. Prompt user and get hourOut  
- while (hourOut < 0 || hourOut > 23)  
{  
 cout << “Invalid input. Please try again.”;  
 cin >> hourOut;  
}

Pseudo Code for GetUserInfo cont…

1. Prompt user and get minuteOut  
   - while (minuteOut < 0 || minuteOut > 59)  
   {   
    cout << “Invalid input. Please try again.”;  
    cin >> minuteIn;  
   }

End of algorithm for GetUserInfo

Flowchart for Time

Start

End

roundedHours = hoursParked

roundedTotal = roundedHours

End

roundedHours = hoursParked + 1  
roundedTotal = roundedHours

If (minutesParked >= 1

If (minuteOut < minuteIn)

minuteOut += 60;  
hourOut -= 1;

hoursParked = hourOut – hourIn  
minutesParked = minuteOut - MinuteIn

Pseudo Code for Time

Start of algorithm for time.

1. If (minuteOut < minuteIn)  
   {  
    minuteOut += 60  
    hourOut -= 1  
    hoursParked = hourOut – hourIn  
    minutesParked = minuteOut – minuteIn  
   }
2. Else   
   {  
    hoursParked = hourOut – hourIn  
    minutesParked = minuteOut – minuteIn  
   }
3. If (minute >= 1)  
   {  
    roundedHours = hoursParked + 1  
    roundedTotal = roundedHours  
   }
4. Else   
   {  
    roundedTotal = hoursParked  
   }

End of algorithm for time.

Flowchart for Rate

Start

End

End

End

Case ‘T’:  
case ‘t’: units = 1  
firstRate = firstTruckRate  
secondRate = secondTruckRate

Case ‘C’:  
case ‘c’: units = 3  
firstRate = firstCarRate  
secondRate = secondCarRate

Case ‘B’:  
case ‘b’: units = 2  
firstRate = firstBusRate  
secondRate = secondBusRate

Pseudo Code for Rate

Start of algorithm for Rate

Switch(vehicle)

1. Case ‘C’:  
   case ‘c’: units = 3  
   firstRate = firstCarRate  
   secondRate = secondCarRate  
   break
2. Case ‘B’:  
   case ‘b’: units = 2  
   firstRate = firstBusRate  
   secondRate = secondBusRate  
   break
3. Case ‘T’:  
   case ‘t’: units = 1

firstRate = firstTruckRate  
secondRate = secondTruckRate  
break

End of algorithm for Rate

Flowchart for Charge

Start

End

End

totalCharge = roundedTotal \* firstRate

totalCharge = (roundedTotal \* firstRate) + ((roundedTotal – units) \* secondRate)

Else

If roundedTotal <= units

Pseudo Code for Charge

Start of algorithm for charge

1. If (roundedTotal <= units)  
   totalCharge = roundedTotal \* firstRate
2. Else  
   totalCharge = (units \* firstRate) + ((roundedTotal – units) \* secondRate)

End of algorithm for charge

Flowchart for printReport

Start

End

Display total charges

Display time parked

Display time out

Display time in

Display type of vehicle

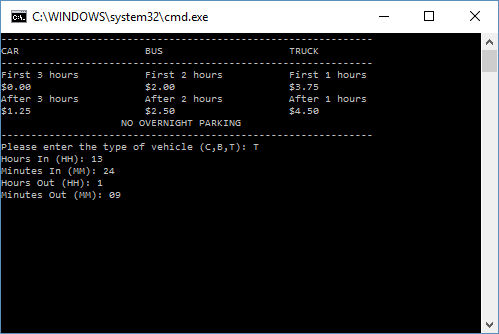
Pseudo Code for printReport

Start of algorithm for printReport

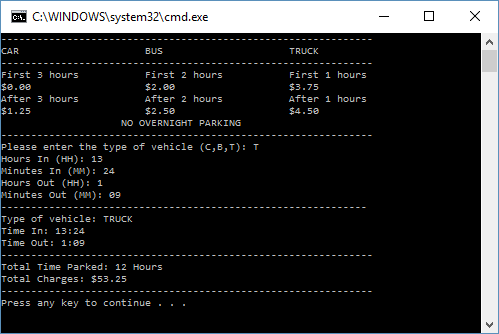
1. Cout << “Type of vehicle” << vehicle
2. Cout << “Time in: “ << (HH) << “:” << (MM)
3. Cout << “Time Out: “ << (HH) << “:” << (MM)
4. Cout << “Total time parked: “ << roundedTotal
5. Cout << “Total Charges: “ << totalCharge

End of algorithm for printReport

**INPUT**



**OUTPUT**



User Instructions

This program will calculate the total cost of using a parking lot given the time entered, the time exited, and the type of vehicle. First enter the type of vehicle by entering either C, B, or T. C for car B for bus and T for truck. Now enter the hours entered in HH form and the same for minutes in MM form. Lastly you will enter the hours exited in HH form and the minutes in MM form. The program will then display the type of vehicle, the time entered and the time exited, and then it will calculate and display the total hours spent at the parking lot and what the total charges were.

Comments

This was probably the toughest lab for me this far. All the concepts that we have learned previously I have picked up and understood pretty quickly but this last chapter that is modular programming has been a bit of a speed bump for me. I feel a lot more comfortable with it now after working with it for so long in this lab trying to get the program to run, but I found myself constantly re-reading things in the book and in the slides to try to understand the fundamentals of modular programming. I’m sure as the semester goes on and we use it more and more I will become more and more comfortable until it is second nature, but I can see how this can be very confusing to a newcomer to coding like me. The transferring of data from one function to another mostly what was tripping me up, but after a lot of trial and error and frustration I have a better understanding and hopefully I will be able to use it well in the upcoming chapters.