/\*----------------------------------------------------------------------------

PROGRAMMER: Name of person

COURSE: CSCI 3133

DATE: 10/12/2009

ASSIGNMENT: Program #2

ENVIRONMENT: Microsoft Vista

FILES INCLUDED: Main, Data, VehicleSelect, ParkingEntries, ParkingHours,

ParkingMins, RndTime, ParkingCharge, PrintAll, and ErrorOut

(.cpp) functions.

PURPOSE: This program accepts User inputs for Parking (hour and

minutes [in/out]), and vehicle type. After acceptable

inputs, the program will output the same information, plus

the total.

INPUT: The User's vehicle type will be input as VehicleParked. The

lot entering times are noted as HourIn and MinIn. And, the

lot exiting times are noted as HourOut and MinOut.

PRECONDITIONS: It is assumed; User will enter the lot before midnight

(24:00 hrs) and will exit the lot no later than midnight,

and User will not enter the lot and leave the lot at the

exact same hour and minute. All program Constants are

located in the function Data.cpp. There are no other

User or Admin Constants in the program that would need

future updating.

OUTPUT: After acceptable User inputs, this program will display the

User's inputs and calculate total hours and charges.

POSTCONDITIONS: Program output calculates total charges rounded up to the

nearest hour. Code cannot predetermine erroneous User

inputs beyond those listed under Errors detected.

ALGORITHM: Prompt User for vehicle type. Record entry. Prompt User for

Parking Lot entry/exit times. Record entries. Calculate

total hours and total minutes. Round to nearest hour.

Assign cost per hour to rounded time. Print User and

programmatic information to screen. Pause until User elects

to exit program.

ERRORS: If User vehicle type is not (C, T, B), an error will occur.

If Parking Lot entry/exit hours is greater than 24 or less

than 0, or if minutes are greater than 60 or less than 0, an

error will occur. If entry time and exit time are exact

same, an error will occur. If exit time is before entry

time, an error will occur. Upon error, User is presented

with a screen notification and program exits after User

elects to exit.

EXAMPLE: User enters: 'C' for Car, Entry hours as 14 and minutes as

23, Exit hours as 18 and minutes as 8. Then, User will be

presented with Time-In and Time-Out confirmation, total

Parking Time (3:45), time rounded to the nearest (greater)

full hour (4:00), and Total Charges ($6.00) based upon

programmed Constants contained in the Data.cpp function.

------------------------------------------------------------------------------

\*/

#include <stdio.h>

char VehicleSelect ();

int ParkingEntries (int \*HourIn, int \*HourOut, int \*MinIn, int \*MinOut);

int ParkingHours (int &HourIn, int &HourOut, int &MinIn, int &MinOut);

int ParkingMins (int &HourIn, int &HourOut, int &MinIn, int &MinOut);

int RndTime (int &TotalHrs, int &TotalMins);

float ParkingCharge (int &Rounded, char &VehicleType);

void PrintAll (char &VehicleType, int &HourIn, int &MinIn, int &HourOut, \

int &MinOut, int &TotalHrs, int &TotalMins, int &Rounded, \

float &ParkingLotCharges);

int main()

{

char VehicleType; /\* User entered either 'C', 'B', or 'T' \*/

int HourIn, HourOut, MinIn, MinOut; /\* User entered Entry and Exit Times \*/

int TotalHrs, TotalMins, Rounded, PkgResults; /\* Program calcs based upon User input \*/

float ParkingLotCharges; /\* Charges use Contants declared in ParkingCharge.cpp and \*/

/\* stored in the Data.cpp function. \*/

VehicleType = VehicleSelect ();

if (VehicleType == 'X') /\* An 'X' indicates that User entered and invalid \*/

{ /\* vehicle entry and the program is about to exit \*/

return(0); /\* via the Errors.cpp function. \*/

}

PkgResults = ParkingEntries(&HourIn, &HourOut, &MinIn, &MinOut);

if (PkgResults == 0) /\* A zero indicates a User time entry error and the \*/

{ /\* program is about to exit via the Errors.cpp function.\*/

return(0);

}

/\* ParkingHours will receive User input and store for multiple functions use \*/

/\* This function utilizes a decrementing For loop to obtain actual hours \*/

TotalHrs = ParkingHours(HourIn, HourOut, MinIn, MinOut);

/\* ParkingMins uses modulo arithmetic to quickly determine actual minutes \*/

/\* Error routine will identify identical entry/exit times which will cause \*/

/\* a program Error exit. \*/

TotalMins = ParkingMins(HourIn, HourOut, MinIn, MinOut);

/\* Total hours will be rounded up to the next hour if total minutes is \*/

/\* greater than zero. \*/

Rounded = RndTime(TotalHrs, TotalMins);

/\* Calculate Lot charges based upon Constant information contained in the \*/

/\* Data.cpp function. Return calculated amount to main function and use \*/

/\* in the PrintAll function. \*/

ParkingLotCharges = ParkingCharge(Rounded, VehicleType);

/\* Gather all User and Program information and display on screen. Include \*/

/\* a pause statement so that screen will remain displayed until User \*/

/\* presses a key to end program. \*/

PrintAll(VehicleType, HourIn, MinIn, HourOut, MinOut, \

TotalHrs, TotalMins, Rounded, ParkingLotCharges);

return(0);

}

/\*---------------------------------------------------------

PURPOSE: ParkingEntries.cpp function will gather

entry/exit time information and return

this information to the main.cpp function

for further processing.

INPUT: User inputs entry time, first in hours

and then in minutes. Next, User inputs

exit time, first in hours and then in

minutes.

OUTPUT: ParkingEntries returns hour and minute

integer values to the main.cpp function.

ALGORITHM: Recieve User inputs and post these to the

main.cpp function.

EXAMPLE: User is queried for "Hour vehicle Entered"

and would enter an integer from 0 to 24.

Next, User is queried for "Minute vehicle

Entered" and would enter an integer from 0

to 60. Next still, User is queried for

"Hour vehicle Left" and again would enter

an integer from 0 to 24. Finally, User is

queried for "Minute vehicle Left" and again

would enter an integer from 0 to 60.

-----------------------------------------------------------

\*/

int ParkingEntries (int \*HourIn, int \*HourOut, int \*MinIn, int \*MinOut)

{

int HourInTemp, MinInTemp, HourOutTemp, MinOutTemp;

/\* Hour entered the parking lot should be within the range allowed \*/

/\* or this function will Error out. \*/

printf("\n Hour vehicle entered lot (0 - 24)? ");

scanf("%d", &HourInTemp);

if ((HourInTemp > 24) || (HourInTemp < 0))

{

ErrorOut();

return(0);

}

\*HourIn = HourInTemp;

/\* Minute entered the parking lot should be within the range allowed \*/

/\* or this function will Error out. \*/

printf(" Minute vehicle entered lot (0 - 60)? ");

scanf("%d", &MinInTemp);

if ((MinInTemp > 60) || (MinInTemp < 0))

{

ErrorOut();

return(0);

}

\*MinIn = MinInTemp;

/\* Hour exited the parking lot should be within the range allowed. \*/

/\* Also, if the HourOut is less than the HourIn, this function will \*/

/\* Error out. \*/

printf("\n Hour vehicle left lot (0 - 24)? ");

scanf("%d", &HourOutTemp);

if ((HourOutTemp > 24) || (HourOutTemp < 0) || (HourOutTemp < HourInTemp))

{

ErrorOut();

return(0);

}

\*HourOut = HourOutTemp;

/\* Minute exited the parking lot should be within the range allowed \*/

/\* or this function will Error out. \*/

printf(" Minute vehicle left lot (0 - 60)? ");

scanf("%d", &MinOutTemp);

if ((MinOutTemp > 60) || (MinOutTemp < 0))

{

ErrorOut();

return(0);

}

\*MinOut = MinOutTemp;

return(1);

}