***Car Parking Lot Project***

***-By Razz Sawhoney***

***About the project:***

***This project is done at the completion of Object Oriented Programming Language Course i.e. C++ in order to test the coverage of the language’s basic syntax , different libraries available in C++,Class and object, operator overloading, file handling, exception handling, etc by the students.***

***Features of this Program:***

1. *This program allows you to handle your car Parking Lot in the best possible way.*
2. *Offers you to keep track of all the required info of a car parked in your lot*
3. *Keeps records of the info like Car Owner’s name,Car Id,entry date and time,exit date and time,and finally calculates the parking charge for you and also the bill the for you.*
4. *Allows you to use auto system date and time and as well as the manual date and time in case you like.*
5. *This program come in two different versions i.e 1st one with the facility of File Handling and the 2nd one without File Handling.*
6. *The 1st version of the program allows you to store your Parking data in the File of your choice on your own system storage directory.*
7. *The 2nd version keeps track of the parking lot position as well i.e. tells you which car is parked in which position in the Lot.*
8. *This program is well secured by Login System so only the concerned personnel can interact with the Parking Data.*
9. *Last but not least the code for this program is well documented. So there is no need to have any further guidelines for the program. But if by chance found anything tough any get comfortably, feel free to contact me.*

***Bugs to be fixed:***

1. *The program is unable to store the time structure data I.e System date and time in a File.*
2. *The login system is yet to be handled by the file handling approach.*
3. *Some particular syntax doesn’t work on Code::Blocks.*
4. *This program is Console based and hence cannot be used as GUI.*

***Flow of Codes:-***

***This program basically consists of six different segments.***

***1st segment(“.h”):*** *This part of the program is actually a header file i.e. of “.h” format and it contains the code for manual date and time entry.*

***2nd segment(“.h”):*** *This part of the program contains Date and Time Conversion algorithm.*

***3rd segment(“.h”):*** *This part of the program contains all the initial declarations for the main class and its different arguments and methods needed for different functionality of the program.*

***4th segment(“.h”):*** *This part of the program contains the definitions for everthings previously declared in the segment 3rd of the program.*

***5th segment(“.cpp”):*** *This part of the program contains the code for the version 1st i.e. with File Handling facility.*

***6th segment(“.cpp”):*** *This final segment of contains the codes for the version 2nd of the program i.e. without File Handling Facility.*

***//Segment 1***

*//#include<bits/stdc++.h>*

*#include <iostream>*

*using namespace std;*

*class TypeError //class to raise error for type exception*

*{*

*char msg[100];*

*public:*

*TypeError(char s[])*

*{*

*strcpy(msg, s);*

*}*

*void showError()*

*{*

*cout << msg << " Please enter valid type" << endl;*

*exit(1);*

*}*

*};*

*class Negative //class to raise error for negative int exception*

*{*

*char msg[100];*

*public:*

*Negative(char s[])*

*{*

*strcpy(msg, s);*

*}*

*void showError()*

*{*

*cout << msg << " Please enter +ve datas!" << endl;*

*return;*

*}*

*};*

*class Invalid //class to raise error for Invalid/unwanted data entry*

*{*

*char msg[100];*

*public:*

*Invalid(char s[])*

*{*

*strcpy(msg, s);*

*}*

*void showError()*

*{*

*cout << msg << " Please enter valid data!" << endl;*

*return;*

*}*

*};*

*class Date*

*{*

*private:*

*int year, month, day;*

*public:*

*Date()*

*{*

*}*

*Date(int netTime)*

*{*

*year = netTime / (24 \* 360);*

*month = (int(netTime) % (24 \* 360)) / (30 \* 24);*

*day = ((int(netTime) % (24 \* 360)) % (30 \* 24)) / 24;*

*}*

*void set\_date()*

*{*

*again:*

*try*

*{*

*cout << "YYYY:MM:DD\n";*

*cin >> year>>month>>day;*

*cin.clear();*

*if (!static\_cast<int>(year) || !static\_cast<int>(month) || !static\_cast<int>(day))*

*throw TypeError("Type Error.");*

*if (year < 0 || month < 0 || day < 0)*

*throw Negative("Negative date entered!!");*

*if (year < 1000 || month > 12 || day > 32)*

*throw Invalid("Invalid date entered!!");*

*}*

*catch (TypeError t)*

*{*

*t.showError();*

*goto again;*

*}*

*catch (Negative n)*

*{*

*n.showError();*

*goto again;*

*}*

*catch (Invalid t)*

*{*

*t.showError();*

*goto again;*

*}*

*}*

*int operator-(Date d2) //To calculate the date difference*

*{*

*Date d;*

*int total\_days;*

*total\_days = (year \* 360 + month \* 30 + day) - (d2.year \* 360 + d2.month \* 30 + d2.day);*

*d.year = total\_days / 360;*

*d.month = (total\_days % 360) / 30;*

*d.day = (total\_days % 360) % 30;*

*return d.year \* 360 \* 24 + d.month \* 30 \* 24 + d.day \* 24;*

*}*

*void displayDate()*

*{*

*char m = ' ', d = ' ';*

*if (month < 10)*

*m = '0';*

*if (day < 10)*

*d = '0';*

*cout << "Date: " << year << " / " << m << month << " / " << d << day << endl;*

*}*

*void displaynetDate()*

*{*

*cout << year << " Years " << month << " Months " << day << " Days" << endl;*

*}*

*};*

*class Time*

*{*

*private:*

*int hour, min, sec;*

*string format;*

*public:*

*Time()*

*{*

*}*

*Time(float netTime)*

*{*

*hour = netTime / 3600;*

*min = (int(netTime) % 3600) / 60;*

*sec = (int(netTime) % 3600) % 60;*

*}*

*void set\_time()*

*{*

*again:*

*try*

*{*

*cout << "HH:MM:SS\n";*

*cin >> hour >> min >> sec;*

*cout << "Enter the Time format AM/PM: ";*

*if (hour < 0 || min < 0 || sec < 0)*

*throw Negative("Negative time entered!!");*

*if (hour > 12 || min > 59 || sec > 59)*

*throw Invalid("Invalid time entered!!");*

*cin.clear();*

*if (!static\_cast<int>(hour) || !static\_cast<int>(min) || !static\_cast<int>(sec))*

*throw TypeError("Type Error.");*

*cin >> format;*

*transform(format.begin(), format.end(), format.begin(), ::toupper); //transforming the lower case characters to upper case*

*}*

*catch (TypeError t)*

*{*

*t.showError();*

*goto again;*

*}*

*catch (Negative n)*

*{*

*n.showError();*

*goto again;*

*}*

*catch (Invalid t)*

*{*

*t.showError();*

*goto again;*

*}*

*}*

*float operator-(Time t2) //To calculate the time difference*

*{*

*Time t;*

*int total\_sec;*

*total\_sec = ((12 + hour) \* 3600 + min \* 60 + sec) - (t2.hour \* 3600 + t2.min \* 60 + t2.sec);*

*t.hour = total\_sec / 3600;*

*t.min = (total\_sec % 3600) / 60;*

*t.sec = (total\_sec % 3600) % 60;*

*if (t2.hour == 12 && t2.format == format)*

*return 12 + t.hour + t.min / 60.0 + t.sec / 3600.0;*

*else if (t2.hour > hour && format == t2.format)*

*return 12 + t.hour + t.min / 60.0 + t.sec / 3600.0;*

*else*

*return t.hour + t.min / 60.0 + t.sec / 3600.0;*

*}*

*void displayTime()*

*{*

*char flag[2];*

*char h = ' ', m = ' ', s = ' ';*

*if (format == "PM")*

*strcpy(flag, "PM");*

*else*

*strcpy(flag, "AM");*

*if (hour < 10)*

*h = '0';*

*if (min < 10)*

*m = '0';*

*if (sec < 10)*

*s = '0';*

*cout << "Time: " << h << hour << ":" << m << min << ":" << s << sec << " " << flag << endl;*

*}*

*void displaynetTime()*

*{*

*cout << hour << " Hours " << min << " Minutes " << sec << " Seconds" << endl;*

*}*

*};*

***// Segment 2 For Date and Time Conversion***

*#include <iostream>*

*using namespace std;*

*class DateTime*

*{*

*private:*

*int year = 0, mon = 0, day = 0, hour, min, sec;*

*public:*

*DateTime()*

*{*

*}*

*DateTime(long tsec)*

*{*

*hour = tsec / 3600;*

*min = (tsec % 3600) / 60;*

*sec = (tsec % 3600) % 60;*

*if (hour > 24)*

*{*

*year = hour / (24 \* 360);*

*mon = (hour % (24 \* 360)) / (24 \* 30);*

*day = ((hour % (24 \* 360)) % (24 \* 30));*

*}*

*}*

*void show()*

*{*

*if (hour > 24)*

*cout << year << " Years " << mon << " months " << day << " days" << endl;*

*cout << hour << " hours " << min << " mins " << sec << " secs" << endl;*

*}*

*};*

***// Segment 3***

***//This is the store house of this program Car Parking***

*#include <ctime> //used for getting local date and time*

*#include "timeConversion.h" //user defined header file to convert date and time in seconds into date and time format*

*#include "date\_time.h" //user defined header file for getting date and time manually*

*class ParkingLotControl*

*{*

*private:*

*//For manual date and time*

*Time entry\_time, exit\_time; //Created two objects of class Time*

*Date entry\_date, exit\_date; //Created two objects of class Date*

*//For system date and time*

*string Entry, Exit;*

*string carId;*

*string name;*

*float Pcharge;*

*public:*

*static int total\_car\_parked;*

*bool entryStatus = false, exitStatus = false;*

*bool isSystem = false; //to work a/c to date and time mode*

*bool usingSystem = false;*

*bool isLotEmpty=true;*

*void get\_entry();*

*void putcarId(string Id);*

*double checkTime();*

*void get\_exit();*

*int checkDate();*

*float CheckTime();*

*void calCharge();*

*void showTotalTime();*

*void showTotalCarParked();*

*void showCharge();*

*void removeCar();*

*void operator++();*

*void operator--();*

*static int getTotalCar();*

*string getCarId();*

*} c; //two class obj created*

***// Segment 4***

*#include <iostream>*

*#include <cstring> //used for string related operations*

*#include <fstream> //used for file handling purpose*

*#include <iomanip> //used for output formatting*

*#include <conio.h> //used for screen clearing purpose*

*#include "CarClass.h" //including the carClass data members*

*const char \*filename = "MeroParkingFinal.dat";*

*string DataBaseUsername="admin";*

*string DataBasePass = "password";*

*time\_t entryT, exitT;*

*using namespace std;*

*void ParkingLotControl::removeCar()*

*{*

*isLotEmpty=true;*

*//entry\_time.reset();*

*//exit\_time.reset();*

*//entry\_date.reset();*

*//exit\_date.reset();*

*carId="None";*

*entryStatus=false;*

*exitStatus=false;*

*}*

*void ParkingLotControl::get\_entry()*

*{*

*int mode;*

*Pcharge = 2; //Min parking charge*

*isLotEmpty=false;*

*cout << "Taking the entry records...\n";*

*cout << "Enter the car owner name: ";*

*cin >> name;*

*up:*

*cout << "Choose Mode:::\n1--->System Date&Time Mode\n2--->Manual Date&Time Mode\n\n";*

*cout << "Choose Date & Time Mode::\t";*

*cin >> mode;*

*if (mode == 1)*

*isSystem = true;*

*else if (mode == 2)*

*isSystem = false;*

*else*

*{*

*cout << "Invalid option selected!!! Try again...\n";*

*goto up;*

*}*

*if (isSystem)*

*{*

*entryT = time(0)+20700; //added to compensate the time loss*

*char \*entryTime = ctime(&entryT);*

*Entry = static\_cast<string>(entryTime);*

*usingSystem = true;*

*}*

*else*

*{*

*cout << "Enter the current date (Year/mon/day): " << endl;*

*entry\_date.set\_date();*

*cout << "Enter the current time: ";*

*entry\_time.set\_time();*

*usingSystem = false;*

*}*

*cout << "Entry records successfully noted..." << endl;*

*}*

*void ParkingLotControl::putcarId(string Id)*

*{*

*carId = Id;*

*}*

*double ParkingLotControl::checkTime()*

*{*

*return difftime(exitT, entryT) / 3600; //returns time diff in hours*

*}*

*void ParkingLotControl::get\_exit()*

*{*

*if (usingSystem == 1)*

*{*

*exitT = time(0)+20700;*

*char \*exitTime = ctime(&exitT);*

*Exit = static\_cast<string>(exitTime);*

*}*

*else*

*{*

*cout << "Taking exit date and time..." << endl;*

*cout << "Enter the current date (Year/mon/day): " << endl;*

*exit\_date.set\_date();*

*cout << "Enter the current time: ";*

*exit\_time.set\_time();*

*}*

*cout << "Exit records successfully noted..." << endl;*

*}*

*int ParkingLotControl::checkDate()*

*{*

*return exit\_date - entry\_date; // returns date in hours*

*}*

*float ParkingLotControl::CheckTime()*

*{*

*return exit\_time - entry\_time; // returns time in hours*

*}*

*void ParkingLotControl::calCharge()*

*{*

*if (usingSystem == 1)*

*goto system;*

*else*

*{*

*float netDate;*

*netDate = checkDate();*

*if (netDate < 0)*

*{*

*cout << "Invalid date entered...." << endl;*

*return;*

*}*

*else if (netDate == 0) //same day*

*{*

*if (CheckTime() < 0)*

*{*

*cout << "Invalid time entered..." << endl;*

*return;*

*}*

*if (CheckTime() >= 3)*

*{*

*Pcharge = Pcharge + (CheckTime() - 3) \* .5;*

*return;*

*}*

*}*

*else if (netDate == 24 && CheckTime() < 24)*

*{*

*Pcharge += (CheckTime() - 3) \* .5;*

*return;*

*}*

*else if (netDate > 24)*

*{*

*Pcharge = (netDate / 24) \* 8;*

*return;*

*}*

*}*

*system:*

*double netTime;*

*netTime = checkTime();*

*if (netTime > 24)*

*{*

*Pcharge = (netTime / 24) \* 8;*

*return;*

*}*

*else if (netTime >= 3)*

*{*

*Pcharge += (netTime - 3) \* .5;*

*return;*

*}*

*}*

*void ParkingLotControl::showTotalTime()*

*{*

*if (usingSystem == 1)*

*{*

*DateTime converted;*

*converted = difftime(exitT, entryT);*

*if (exitStatus == 1)*

*converted.show();*

*else*

*cout << "Exit records not taken yet..." << endl;*

*return;*

*}*

*else*

*{*

*float netDate = checkDate();*

*if (netDate < 0 || exitStatus == false)*

*{*

*cout << "Exit records not taken yet..." << endl;*

*return;*

*}*

*else if (netDate == 0) //Car exit on the same day*

*{*

*Time t;*

*t = CheckTime() \* 3600; // converting basic type to class type*

*t.displaynetTime();*

*return;*

*}*

*else if (netDate == 24 && CheckTime() < 24)*

*{*

*Time t;*

*t = CheckTime() \* 3600; // converting basic type to class type*

*t.displaynetTime();*

*return;*

*}*

*else*

*{*

*Date d;*

*d = checkDate(); // converting basic type to class type*

*d.displaynetDate();*

*return;*

*}*

*}*

*}*

*void ParkingLotControl::showTotalCarParked()*

*{*

*cout << "\n~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~\n";*

*cout << "\n\tEntry records and more info:" << endl;*

*cout << "\n\tCar Owner: " << name << endl;*

*cout << "\n\tCarID: " << carId << endl;*

*if (usingSystem == 1)*

*{*

*cout << "\n\tEntry date & time: \n";*

*cout << "\n\t";*

*cout << Entry << endl;*

*cout << "\n\tExit date & time: \n";*

*cout << "\n\t";*

*if (exitStatus == 1)*

*cout << Exit << endl;*

*cout << "\n\tTotal time of staying==";*

*showTotalTime();*

*}*

*else*

*{*

*cout << "\n\tEntry date & time: \n";*

*cout << "\n\t";*

*entry\_date.displayDate();*

*cout << "\n\t";*

*entry\_time.displayTime();*

*cout << "\n\tExit date & time: \n";*

*if (exitStatus == 1)*

*{*

*cout << "\n\t";*

*exit\_date.displayDate();*

*cout << "\n\t";*

*exit\_time.displayTime();*

*}*

*cout << "\n\tTotal time of staying==";*

*showTotalTime();*

*}*

*cout << "\n~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~\n";*

*}*

*void ParkingLotControl::showCharge()*

*{*

*if (entryStatus == 1 && exitStatus == 1)*

*{*

*calCharge();*

*cout << "Exit reports and more info:" << endl;*

*cout << "Car Owner: " << name << endl;*

*cout << "CarID: " << carId << endl;*

*cout << "Entry date & time: \n";*

*if (isSystem == 1)*

*cout << Entry << endl;*

*else*

*{*

*entry\_date.displayDate();*

*entry\_time.displayTime();*

*}*

*cout << "Exit date & time:\n";*

*if (isSystem == 1)*

*cout << Exit << endl;*

*else*

*{*

*exit\_date.displayDate();*

*exit\_time.displayTime();*

*}*

*cout << endl*

*<< "Total staying time: " << endl;*

*showTotalTime();*

*cout << "Total Parking charge: "*

*<< " $ " << Pcharge << endl;*

*cout << "Have a safe journey ahead...Happy Parking....!!!" << endl;*

*}*

*else*

*cout << "Sir please departure the car first..." << endl;*

*if (total\_car\_parked < 0)*

*cout << "There's no any car parked in our Lot...." << endl;*

*}*

*void ParkingLotControl::operator++()*

*{*

*++total\_car\_parked;*

*}*

*void ParkingLotControl::operator--()*

*{*

*--total\_car\_parked;*

*}*

*int ParkingLotControl::getTotalCar()*

*{*

*return total\_car\_parked;*

*}*

*string ParkingLotControl::getCarId()*

*{*

*return carId;*

*}*

*int ParkingLotControl::total\_car\_parked=0;*

*void delete\_record()*

*{*

*int found = 0;*

*string carNo;*

*cout << "\n\tEnter the car no you want to get departured : ";*

*cin >> carNo;*

*ifstream inFile;*

*inFile.open(filename, ios::binary | ios::in);*

*ofstream outFile;*

*outFile.open("temp1.dat", ios::out | ios::binary);*

*outFile.seekp(0, ios::beg);*

*inFile.seekg(0, ios::beg);*

*while (inFile.read((char \*)&c, sizeof(c)))*

*{*

*if (c.getCarId() == carNo)*

*{*

*found = 1;*

*break;*

*}*

*}*

*if (found == 1)*

*{*

*if (c.getCarId() == carNo && c.exitStatus == false)*

*{*

*cout << "This car is not Departured from our DataBase yet...!!" << endl;*

*return;*

*}*

*else*

*{*

*inFile.seekg(0, ios::beg);*

*outFile.seekp(0, ios::beg);*

*while (inFile.read((char \*)&c, sizeof(c)))*

*{*

*if (c.getCarId() != carNo)*

*outFile.write((char \*)&c, sizeof(c));*

*if (c.getCarId() == carNo && c.exitStatus == 1)*

*{*

*c.showCharge();*

*cout << "\n\t Car with Id " << carNo << " is successfully departured from our DateBase...\n\n";*

*}*

*}*

*inFile.close();*

*outFile.close();*

*remove(filename);*

*rename("temp1.dat", filename);*

*}*

*}*

*else*

*cout << "Car with this ID doesn't exist in our DataBase!!!\n\n";*

*}*

*void menu()*

*{*

*cout << setfill('=') << setw(100);*

*cout << "\n\t\t\*\*\*CAR PARKING DATA BASE\*\*\*";*

*cout << setfill('=') << setw(100);*

*cout << setfill('~') << setw(100) << endl;*

*cout << "\n\n\t\t1-->Do entry the car" << endl;*

*cout << "\t\t2-->Take departure" << endl;*

*cout << "\t\t3-->Print the bill" << endl;*

*cout << "\t\t4-->Info on all parked Cars" << endl;*

*cout << "\t\t5-->Log Out" << endl*

*<< endl;*

*cout << setfill('~') << setw(100) << endl;*

*cout << "\n\t\tEnter your choice here==\t";*

*}*

*void LoginPanel()*

*{*

*cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;*

*cout << setw(43) << "\*\*\*WEL-COME To RAZZZ Car Parking Lot\*\*\*" << endl;*

*cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl*

*<< endl;*

*cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl*

*<< endl;*

*cout << "~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~" << endl;*

*cout << "1->Login to your ParkingLot DataBase..." << endl;*

*cout << "2->Change Your Password..." << endl;*

*cout << "3->Exit from the Program" << endl;*

*cout << "~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~" << endl;*

*cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;*

*}*

*bool checkLogin(string username,string pass)*

*{*

*if(DataBaseUsername==username && DataBasePass==pass)*

*return true;*

*else*

*return false;*

*}*

*void changePass(string pass)*

*{*

*DataBasePass = pass;*

*}*

*void changeUsername(string user)*

*{*

*DataBaseUsername=user;*

*}*

***// Segment 5. Version 1.0 :: Using File handling***

***//Progam 8. A parking garage charges a $2.00 minimum fee to park for up to three hours. The garage charges an additional $0.50 per hour for each hour or part thereof in excess of three hours. People who park their cars for longer than 24 hours will pay $8.00 per day. Write a program that calculates and prints the parking charges. The inputs to your program are the date and time when a car enters the parking garage, and the date and time when the same car leaves the parking garage. Keep track of number of cars in the garage overloading ++ and -- operators****.*

*#include "CarParkingControl.h" //user defined header which includes all the required methods and functionality*

*int main()*

*{*

*char option, option1;*

*bool isLoginned = false;*

*bool firstlog = true;*

*loginPanel:*

*while (1)*

*{*

*LoginPanel();*

*if (firstlog)*

*{*

*cout << "NOTE: Your default username='admin' & password='password'. Please change it for more security reason..." << endl;*

*cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;*

*}*

*cin >> option1;*

*switch (option1)*

*{*

*case '1':*

*{*

*enterAgain:*

*string userNm,password;*

*cout<<"\nPlease enter the usename==\t";*

*cin>>userNm;*

*cout << "\nPlease enter your login password==\t";*

*getch();*

*char ch1;*

*ch1 = \_getch();*

*//while (ch1!=13) //Works on Windows where 13 is ASCII value of Enter key*

*while(ch1!='\n')*

*{*

*password+=ch1;*

*cout << '\*';*

*ch1 = \_getch();*

*}*

*if (checkLogin(userNm,password))*

*{*

*cout << "\n\tSuccessfully loginned.....\n\n" << endl;*

*isLoginned = true;*

*goto DataBase;*

*}*

*else*

*{*

*cout << "\n\tUsename or password not matched!!!!Please try again..." << endl;*

*isLoginned = false;*

*goto enterAgain;*

*}*

*break;*

*}*

*case '2':*

*{*

*enterAgain2:*

*string tempPass="", newPass="";*

*cout << "\nSir Please enter your current password==\t";*

*getch(); //pause the program for a while*

*char ch2;*

*ch2 = \_getch();*

*//while (ch2!=13) //Works on Windows where 13 is ASCII value of Enter key*

*while(ch2!='\n')*

*{*

*tempPass+=ch2;*

*cout << '\*';*

*ch2 = \_getch();*

*}*

*if (DataBasePass==tempPass)*

*{*

*cout << "\nEnter your new and a bit stronger password==\t" << endl;*

*getch();*

*char ch3;*

*ch3 = \_getch();*

*//while (ch3!=13) //Works on Windows where 13 is ASCII value for Enter key*

*while(ch3!='\n')*

*{*

*newPass+=ch3;*

*cout << '\*';*

*ch3 = \_getch();*

*}*

*changePass(newPass);*

*char choice;*

*string temUser;*

*cout << "\n\t\tPassword successfully changed..." << endl;*

*firstlog = false;*

*cout<<"Do you wish to change the username as well????[Y/N]";*

*cin>>choice;*

*if(choice=='Y' || choice=='y')*

*{*

*cout<<"Enter your new username...:\t";*

*cin>>temUser;*

*changeUsername(temUser);*

*cout<<"Username successfully changed..!!\n";*

*}*

*}*

*else*

*{*

*cout << "\n\t\tWrong password!!! Please Try again..." << endl;*

*goto enterAgain2;*

*}*

*break;*

*}*

*case '3':*

*char confirm1[2];*

*cout << "Do you really wanna exit (Y/N)?? ";*

*cin >> confirm1;*

*if (strcmp(confirm1, "Y") == 0 || strcmp(confirm1, "y") == 0)*

*{*

*cout << "Thanks for visiting us...\n";*

*goto last;*

*}*

*else*

*break;*

*default:*

*cout << "Invalid option selected...\n";*

*break;*

*}*

*}*

*DataBase:*

*while (isLoginned)*

*{*

*menu();*

*cin >> option;*

*switch (option)*

*{*

*case '1':*

*{*

*string Id;*

*bool ifexist;*

*cout<<"Enter the carId:\t";*

*cin>>Id;*

*fstream file;*

*file.open(filename,ios::in|ios::out|ios::binary|ios::app);*

*file.seekg(0,ios::beg);*

*while (file.read((char \*)&c, sizeof(c)))*

*{*

*if (c.getCarId()== Id)*

*{*

*ifexist=true;*

*break;*

*}*

*else*

*ifexist=false;*

*}*

*if(ifexist)*

*{*

*cout<<"Car with the ID "<<Id<<" already exist in our DataBase!!!"<<endl;*

*break;*

*}*

*else*

*{*

*c.putcarId(Id);*

*c.entryStatus = true;*

*++c;*

*c.get\_entry();*

*file.clear();*

*file.seekp(0,ios::end);*

*file.write((char \*)&c, sizeof(c));*

*cout << "Data is stored..." << endl;*

*file.close();*

*}*

*break;*

*}*

*case '2':*

*{*

*string id;*

*int recordFound=0;*

*cout << "Enter the CarId: ";*

*cin >> id;*

*fstream file;*

*file.open(filename, ios::in|ios::binary);*

*file.clear();*

*file.seekg(0,ios::beg);*

*while (file.read((char \*)&c, sizeof(c)))*

*{*

*if (c.getCarId()== id)*

*{*

*recordFound=1;*

*break;*

*}*

*}*

*file.close();*

*if(recordFound==1)*

*{*

*ofstream fout;*

*fout.open("temp1.dat", ios::out | ios::binary);*

*fout.seekp(0,ios::beg);*

*ifstream infile(filename, ios::binary | ios::in);*

*while (infile.read((char \*)&c, sizeof(c)))*

*{*

*if (c.getCarId()==id)*

*{*

*c.get\_exit();*

*c.exitStatus = true;*

*}*

*fout.write((char \*)&c, sizeof(c));*

*}*

*infile.close();*

*fout.close();*

*remove(filename);*

*rename("temp1.dat",filename);*

*}*

*else*

*cout << "Sorry sir!!! There is no car with this ID in our Parking Lot!!!" << endl;*

*break;*

*}*

*case '3':*

*{*

*delete\_record();*

*break;*

*}*

*case '4':*

*{*

*cout << "\n\t=== View the Records in the Parking Database ===\n";*

*ifstream infile(filename, ios::binary | ios::in);*

*int d = 1;*

*while (infile.read((char \*)&c, sizeof(c)))*

*{*

*infile.clear();*

*//system("pause");*

*cout << "\n\tThe Car position in the Parking Lot : " << d << endl;*

*d++;*

*c.showTotalCarParked();*

*}*

*cout << "\n\tTotal no of cars in the lot: " <<d-1<< endl;*

*if(d==1)*

*cout<<"\n\tNo any records found in our DataBase "<<endl;*

*infile.close();*

*break;*

*}*

*case '5':*

*{*

*char confirm[1];*

*cout << "Do you really wanna sign out (Y/N)?? ";*

*cin >> confirm;*

*if (strcmp(confirm, "Y") == 0 || strcmp(confirm, "y") == 0)*

*{*

*cout << "Thanks for visiting us...\n";*

*goto loginPanel;*

*}*

*else*

*break;*

*}*

*default:*

*cout << "Invalid option selected...\n";*

*break;*

*}*

*}*

*last:*

*return 0;*

*}*

***// Segment 6 Version 2.0 :: Without using File***

*#include "CarParkingControl.h"*

*#define parkingSize 10*

*int main()*

*{*

*ParkingLotControl p[parkingSize];*

*int carNo;*

*int displayCount=0;*

*int recentlydeleted;*

*char option, option1;*

*bool isLoginned = false;*

*bool recentDeletion=false;*

*bool firstlog = true;*

*loginPanel2:*

*while(1)*

*{*

*displayCount++;*

*if(displayCount>8)*

*{*

*clrscr();*

*displayCount=0;*

*}*

*LoginPanel();*

*if (firstlog)*

*{*

*cout << "NOTE: Your default username: admin\npassword: 'password' \nPlease change it for more security reason..." << endl;*

*cout << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;*

*}*

*cin >> option1;*

*switch (option1)*

*{*

*case '1':*

*{*

*enterAgain3:*

*string userNm;*

*string password;*

*cout<<"\nPlease enter the usename==\t";*

*cin>>userNm;*

*cout << "\nPlease enter your login password==\t";*

*getch();*

*char ch1;*

*ch1 = \_getch();*

*//while (ch1!=13) //Works on Windows*

*while(ch1!='\n')*

*{*

*password+=ch1;*

*cout << '\*';*

*ch1 = \_getch();*

*}*

*if (checkLogin(userNm,password))*

*{*

*cout << "\n\tSuccessfully loginned.....\n\n" << endl;*

*isLoginned = true;*

*goto DataBase2;*

*}*

*else*

*{*

*cout << "\n\tUsename or password not matched!!!!Please try again..." << endl;*

*isLoginned = false;*

*goto enterAgain3;*

*}*

*break;*

*}*

*case '2':*

*{*

*enterAgain4:*

*string tempPass="", newPass="";*

*cout << "\nSir Please enter your current password==\t";*

*getch(); //pause the program for a while*

*char ch2;*

*ch2 = \_getch();*

*//while (ch2!=13) //Works on Windows*

*while(ch2!='\n')*

*{*

*tempPass+=ch2;*

*cout << '\*';*

*ch2 = \_getch();*

*}*

*if (DataBasePass==tempPass)*

*{*

*cout << "\nEnter your new and a bit stronger password==\t" << endl;*

*getch();*

*char ch3;*

*ch3 = \_getch();*

*//while (ch3!=13) //Works on Windows*

*while(ch3!='\n')*

*{*

*newPass+=ch3;*

*cout << '\*';*

*ch3 = \_getch();*

*}*

*changePass(newPass);*

*char choice;*

*string temUser;*

*cout << "\n\t\tPassword successfully changed..." << endl;*

*firstlog = false;*

*cout<<"Do you wish to change the username as well????[Y/N]";*

*cin>>choice;*

*if(choice=='Y' || choice=='y')*

*{*

*cout<<"Enter your new username...:\t";*

*cin>>temUser;*

*changeUsername(temUser);*

*cout<<"Username successfully changed..!!\n";*

*}*

*}*

*else*

*{*

*cout << "\n\t\tWrong password!!! Please Try again..." << endl;*

*goto enterAgain4;*

*}*

*break;*

*}*

*case '3':*

*char confirm1[2];*

*cout << "Do you really wanna exit (Y/N)?? ";*

*cin >> confirm1;*

*if (strcmp(confirm1, "Y") == 0 || strcmp(confirm1, "y") == 0)*

*{*

*cout << "Thanks for visiting us...\n";*

*goto last;*

*}*

*else*

*break;*

*default:*

*cout << "Invalid option selected...\n";*

*break;*

*}*

*}*

*DataBase2:*

*while (isLoginned)*

*{*

*displayCount++;*

*if(displayCount>9)*

*{*

*clrscr();*

*displayCount=0;*

*}*

*carNo=ParkingLotControl::getTotalCar();*

*//cout<<"Total car in lot="<<carNo<<endl;*

*menu();*

*cin >> option;*

*switch (option)*

*{*

*case '1':*

*{*

*if(carNo==parkingSize)*

*{*

*cout<<"Sorry!! The parking lot is completely FULL...Plz Try after sometime\n";*

*break;*

*}*

*string Id;*

*bool ifexist=false;*

*cout<<"Enter the carId:\t";*

*cin>>Id;*

*if(carNo>0)*

*{*

*for(int j=0;j<carNo;j++)*

*{*

*if (p[j].getCarId()== Id)*

*{*

*ifexist=true;*

*break;*

*}*

*}*

*}*

*else*

*goto enterData;*

*if(ifexist)*

*{*

*cout<<"Car with the ID "<<Id<<" already exist in our DataBase!!!"<<endl;*

*break;*

*}*

*else*

*{*

*enterData:*

*if(recentDeletion)*

*{*

*p[recentlydeleted].putcarId(Id);*

*p[recentlydeleted].entryStatus = true;*

*++p[recentlydeleted]; //increases the car num by 1*

*p[recentlydeleted].get\_entry();*

*carNo++; //Increasing the index for more car entry*

*recentDeletion=false;*

*}*

*else*

*{*

*p[carNo].putcarId(Id);*

*p[carNo].entryStatus = true;*

*++p[carNo]; //increases the car num by 1*

*p[carNo].get\_entry();*

*carNo++; //Increasing the index for more car entry*

*}*

*}*

*break;*

*}*

*case '2': //Car Id is matched and is ready to get departured*

*{*

*string id;*

*int recordNo;*

*int recordFound=0;*

*cout << "Enter the CarId: ";*

*cin >> id;*

*for(int i=0;i<carNo;i++)*

*{*

*if (p[i].getCarId()== id)*

*{*

*recordFound=1;*

*recordNo=i;*

*break;*

*}*

*}*

*if(recordFound==1)*

*{*

*p[recordNo].get\_exit();*

*p[recordNo].exitStatus = true;*

*}*

*else*

*cout << "Sorry sir!!! There is no car with this ID in our Parking Lot!!!" << endl;*

*break;*

*}*

*case '3': //Delete the particular car from the dataBase and display the charges for the departured one*

*{*

*string id;*

*int recordNo;*

*int recordFound=0;*

*cout << "Enter the CarId: ";*

*cin >> id;*

*for(int i=0;i<carNo;i++)*

*{*

*if (p[i].getCarId()== id)*

*{*

*recordFound=1;*

*recordNo=i;*

*break;*

*}*

*}*

*if(recordFound==1 && p[recordNo].exitStatus==true)*

*{*

*p[recordNo].showCharge();*

*--p[recordNo]; //decreases the car num by 1*

*for(int i=0;i<carNo;i++)*

*{*

*if(i==recordNo)*

*{*

*p[recordNo].removeCar();*

*recentlydeleted=recordNo;*

*recentDeletion=true;*

*}*

*/\**

*if(i==recordNo)*

*{*

*for(int j=i;j<carNo-1;j++)*

*p[j]=p[j+1]; //stores all other car except that one*

*break;*

*}*

*\*/*

*}*

*}*

*else if(recordFound==1 && p[recordNo].exitStatus==false)*

*cout<<"This car with id "<<id<<" is still to take exit..."<<endl;*

*else if(recordFound==0)*

*cout<<"Car with Id "<<id<<" is not found in our DataBase!!"<<endl;*

*break;*

*}*

*case '4':*

*{*

*cout << "\n\t=== View the Records in the Parking Database ===";*

*cout << "\n";*

*int i=0; //since used outside the for loop*

*for(i=0;i<parkingSize;i++)*

*{*

*if(p[i].isLotEmpty)*

*cout<<"\n\t The Lot no. "<<i+1<<" is empty!!!"<<endl;*

*else if(!p[i].isLotEmpty)*

*{*

*cout << "\n\tThe Car position in the Parking Lot : " << i+1 << endl;*

*p[i].showTotalCarParked();*

*}*

*}*

*if(i==0)*

*cout<<"\n\tNo any records found in our DataBase "<<endl;*

*break;*

*}*

*case '5':*

*{*

*char confirm[1];*

*cout << "Do you really wanna sign out (Y/N)?? ";*

*cin >> confirm;*

*if (strcmp(confirm, "Y") == 0 || strcmp(confirm, "y") == 0)*

*{*

*cout << "Thanks for visiting us...\n";*

*goto loginPanel2;*

*}*

*else*

*break;*

*}*

*default:*

*cout << "Invalid option selected...\n";*

*break;*

*}*

*}*

*last:*

*return 0;*

*}*

***That’s the end of the program!!!***

***For further info. Contact me rajsahani1819@gmail.com***

***THANKS!!!***