***Car Parking Lot Project***

***-By IUCRAZZ ShaNidhi***

***About the project:***

***This project is done as a part of Object Oriented Programming Language Course i.e. C++ in order to test the students about 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 with good Console based User Interface.*
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.*

***Note: For Program with File Handling look to my other repos which is without Graphics Feature.***

1. *This program is well secured by Login System so only the concerned personnel can interact with the Parking Data.*
2. *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 but boosted with Graphics Features.*

***Flow of Codes:-***

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

***1st segment(“date\_time.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(“timeCoversion.h”):*** *This part of the program contains Date and Time Conversion algorithm.*

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

***5th segment(“AddGraphics.h”):*** *This part of the Project is responsible for adding graphics functionality and features to the program like full screen console,text colors,text scrolling etc.*

***6th segment(“carParkingLotWithoutFile.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>*

*#include<cstring>*

*#include "Addgraphics.h"*

*bool isNumber(string);*

*using namespace std;*

*class TypeError*

*{*

*char msg[100];*

*public:*

*TypeError(char s[])*

*{*

*strcpy(msg, s);*

*}*

*void showError()*

*{*

*setColor(yellow,black);*

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

*// exit(1);*

*}*

*};*

*class Negative*

*{*

*char msg[100];*

*public:*

*Negative(char s[])*

*{*

*strcpy(msg, s);*

*}*

*void showError()*

*{*

*setColor(yellow,black);*

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

*return;*

*}*

*};*

*class Invalid*

*{*

*char msg[100];*

*public:*

*Invalid(char s[])*

*{*

*strcpy(msg, s);*

*}*

*void showError()*

*{*

*setColor(yellow,black);*

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

*return;*

*}*

*};*

*class Date*

*{*

*private:*

*int year, month, day;*

*string syear,smonth,sday;*

*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*

*{*

*setColor(black,yellow);*

*cout << "In YYYY:MM:DD Format Only Please!\n";*

*setColor(white,black);*

*cin.clear();*

*getline(cin,syear,':');*

*getline(cin,smonth,':');*

*getline(cin,sday);*

*if (!isNumber(syear) || !isNumber(smonth) || !isNumber(sday))*

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

*year=atoi(syear.c\_str());*

*month=atoi(smonth.c\_str());*

*day=atoi(sday.c\_str());*

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

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

*if (year < 1000 || year > 2020 || 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)*

*{*

*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';*

*setColor(green,black);*

*cout << "\n\tDate: " << year << " / " << m << month << " / " << d << day << endl;*

*}*

*void displaynetDate()*

*{*

*setColor(green,black);*

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

*}*

*};*

*class Time*

*{*

*private:*

*int hour, min, sec;*

*string hr\_s, min\_s, sec\_s;*

*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*

*{*

*setColor(black,yellow);*

*cin.clear();*

*cout << "\nIn HH:MM:SS & 12 Hours Format Only Please!\n";*

*setColor(white,black);*

*getline(cin, hr\_s, ':');*

*getline(cin, min\_s, ':');*

*getline(cin, sec\_s);*

*if (!isNumber(hr\_s) || !isNumber(min\_s) || !isNumber(sec\_s))*

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

*hour=atoi(hr\_s.c\_str());*

*min=atoi(min\_s.c\_str());*

*sec=atoi(sec\_s.c\_str());*

*tmformat:*

*setColor(green,black);*

*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!!");*

*setColor(white,black);*

*cin >> format;*

*if(format == "pm" || format == "PM" || format == "am" || format == "AM")*

*{}*

*else*

*goto tmformat;*

*transform(format.begin(), format.end(), format.begin(), ::toupper);*

*}*

*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)*

*{*

*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';*

*setColor(green,black);*

*cout << "\n\tTime: " << h << hour << ":" << m << min << ":" << s << sec << " " << flag << endl;*

*}*

*void displaynetTime()*

*{*

*setColor(green,black);*

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

*}*

*};*

*bool isNumber(string s)*

*{*

*for (int i = 0; i < s.length(); i++)*

*if (isdigit(s[i]) == false)*

*return false;*

*return true;*

*}*

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

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

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

*#include "CarClass.h"*

*#include<chrono>*

*#include<string>*

*const char \*filename = "MeroParkingFinal1.txt"; //For File system project*

*string DataBaseUsername="iucrazz";*

*string DataBasePass = "iucrazz123";*

*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;*

*setColor(pink,lightblue);*

*animateTxt("Taking the entry records...\n");*

*setColor(green,black);*

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

*setColor(white,black);*

*cin>>name;*

*up:*

*setColor(pink,lightblue);*

*animateTxt("Choose Mode:::");*

*setColor(yellow,black);*

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

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

*setColor(white,black);*

*cin >> mode;*

*if (mode == 1)*

*isSystem = true;*

*else if (mode == 2)*

*isSystem = false;*

*else*

*{*

*setColor(yellow,black);*

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

*goto up;*

*}*

*if (isSystem)*

*{*

*entryT = time(0); //add 20700 to compensate the time loss in case needed*

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

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

*usingSystem = true;*

*}*

*else*

*{*

*setColor(green,black);*

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

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

*setColor(green,black);*

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

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

*usingSystem = false;*

*}*

*setColor(white,pink);*

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

*}*

*void ParkingLotControl::get\_exit()*

*{*

*if (usingSystem == 1)*

*{*

*exitT = time(0); //add 20700 to compensate the time loss in case needed*

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

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

*}*

*else*

*{*

*setColor(pink,lightblue);*

*animateTxt("Taking exit date and time...\n");*

*enterDate:*

*setColor(green,black);*

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

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

*float netDate = checkDate();*

*if (netDate < 0)*

*{*

*setColor(red,black);*

*cout<<"Invalid Date provided!!! Exit Date cannot be before Entry Date i.e must be (Exit\_date > Entry\_date)..."<<endl;*

*goto enterDate;*

*}*

*setColor(green,black);*

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

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

*}*

*setColor(white,pink);*

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

*}*

*void ParkingLotControl::putcarId(string Id)*

*{*

*carId = Id;*

*}*

*double ParkingLotControl::checkTime()*

*{*

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

*}*

*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)*

*{*

*setColor(yellow,black);*

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

*return;*

*}*

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

*{*

*if (CheckTime() < 0)*

*{*

*setColor(yellow,black);*

*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)*

*{*

*setColor(green,black);*

*converted.show();*

*}*

*else*

*{*

*setColor(yellow,black);*

*cout << "\tExit records not taken yet..." << endl;*

*}*

*return;*

*}*

*else*

*{*

*float netDate = checkDate();*

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

*{*

*setColor(yellow,black);*

*cout << "\tExit 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()*

*{*

*setColor(red,lightblue);*

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

*setColor(white,pink);*

*animateTxt("\n\tEntry records and more info:\t");*

*setColor(green,black);*

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

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

*if (usingSystem == 1)*

*{*

*setColor(white,pink);*

*animateTxt("\n\tEntry date & time:\t \n");*

*setColor(green,black);*

*cout <<endl<< Entry << endl;*

*setColor(white,pink);*

*animateTxt("\n\tExit date & time:\t \n");*

*if (exitStatus == 1)*

*{*

*setColor(green,black);*

*cout <<endl<< Exit << endl;*

*}*

*else{*

*setColor(yellow,black);*

*cout<<"Exit records not taken so far!!!"<<endl;*

*}*

*setColor(white,pink);*

*animateTxt("\n\tTotal time of staying==\t");*

*showTotalTime();*

*}*

*else*

*{*

*setColor(white,pink);*

*animateTxt("\n\tEntry date & time:\t \n");*

*entry\_date.displayDate();*

*entry\_time.displayTime();*

*setColor(white,pink);*

*animateTxt("\n\tExit date & time:\t \n");*

*if (exitStatus == 1)*

*{*

*setColor(green,black);*

*exit\_date.displayDate();*

*exit\_time.displayTime();*

*}*

*else{*

*setColor(yellow,black);*

*cout<<"Exit records not taken so far!!!"<<endl;*

*}*

*setColor(white,pink);*

*animateTxt("\n\tTotal time of staying==\t");*

*showTotalTime();*

*}*

*setColor(red,lightblue);*

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

*}*

*void ParkingLotControl::showCharge()*

*{*

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

*{*

*calCharge();*

*setColor(black,yellow);*

*animateTxt("Exit reports and more info:\t\n");*

*setColor(green,black);*

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

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

*setColor(black,yellow);*

*animateTxt("Entry date & time:\t \n");*

*if (isSystem == 1)*

*{*

*setColor(green,black);*

*cout << Entry << endl;*

*}*

*else*

*{*

*entry\_date.displayDate();*

*entry\_time.displayTime();*

*}*

*setColor(black,yellow);*

*animateTxt("Exit date & time:\t\n");*

*if (isSystem == 1)*

*{*

*setColor(green,black);*

*cout << Exit << endl;*

*}*

*else*

*{*

*exit\_date.displayDate();*

*exit\_time.displayTime();*

*}*

*setColor(black,yellow);*

*animateTxt("\nTotal staying time: \t\n");*

*showTotalTime();*

*setColor(black,yellow);*

*cout<<"\nTotal Parking charge: ";*

*setColor(green,black);*

*cout<<" $"<<Pcharge<<" only."<<endl;*

*setColor(pink,lightblue);*

*cout << "\nHave a safe journey ahead...Happy TO Serve....!!!" << endl;*

*}*

*else*

*{*

*setColor(yellow,black);*

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

*}*

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

*{*

*setColor(yellow,black);*

*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;*

*setColor(green,black);*

*cout << "\n\tEnter the car Id you want to get departured : ";*

*setColor(white,black);*

*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)*

*{*

*setColor(yellow,black);*

*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();*

*setColor(black,yellow);*

*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*

*{*

*setColor(yellow,black);*

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

*}*

*}*

*void LoginPanel()*

*{*

*setColor(red,black);*

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

*setColor(white,blue);*

*animateTxt("\t\t\*\*\*WEL-COME To IUCRAZZ Car Parking Lot\*\*\* \t\t");*

*setColor(yellow,black);*

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

*setColor(black,green);*

*cout<<"Administered By: IUCRazz ShaNidhi";*

*setColor(red,black);*

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

*setColor(red,black);*

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

*<< endl;*

*setColor(yellow,black);*

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

*setColor(green,black);*

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

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

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

*setColor(yellow,black);*

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

*setColor(red,black);*

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

*}*

*void menu()*

*{*

*setColor(yellow,black);*

*cout << "===================================================================="<<endl<<endl;*

*setColor(white,blue);*

*animateTxt("\t\t\*\*\*CAR PARKING DATA BASE\*\*\*\t\t");*

*setColor(yellow,black);*

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

*setColor(black,green);*

*cout<<"Administered By: IUCRazz ShaNidhi";*

*setColor(red,black);*

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

*setColor(yellow,black);*

*cout <<endl<<endl<< "===================================================================="<<endl;*

*setColor(pink,black);*

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

*setColor(green,black);*

*cout << "\n\n\t\t1-->Park the Car" << endl;*

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

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

*cout << "\t\t4-->Display All Parked Cars" << endl;*

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

*setColor(pink,black);*

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

*setColor(green,black);*

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

*}*

*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;*

*}*

*string InputPass()*

*{*

*char ch1;*

*string password;*

*while (ch1 != 13)*

*{*

*ch1 = \_getch();*

*if(ch1 == '\b' && password.length())*

*{*

*cout<<'\b'<<' '<<'\b';*

*password.erase( password.end() -1 );*

*}*

*else if(ch1 != 13){*

*setColor(white,black);*

*cout << '\*';*

*password.push\_back(ch1);*

*}*

*}*

*return password;*

*}*

***// Segment 5. For Graphics Functionality and Features***

*#define WIN32\_LEAN\_AND\_MEAN*

*#include <windows.h>*

*#include <stdio.h>*

*//Colors Defn*

*unsigned int red =4;*

*unsigned int blue =1;*

*unsigned int green =10;*

*unsigned int yellow =6;*

*unsigned int white =15;*

*unsigned int gray =8;*

*unsigned int pink =5;*

*unsigned int black =32;*

*unsigned int lightblue =3;*

*unsigned int lightyellow =14;*

*unsigned int lightgreen =10;*

*unsigned int lightred =36;*

*void setColor(unsigned int fg\_Color, unsigned int bg\_Color)*

*{*

*HANDLE hStdOut;*

*hStdOut = GetStdHandle(STD\_OUTPUT\_HANDLE);*

*if (hStdOut == INVALID\_HANDLE\_VALUE) {*

*return;*

*}*

*SetConsoleTextAttribute(hStdOut, (bg\_Color << 4) | fg\_Color);*

*}*

*void con\_def\_Color(void)*

*{*

*HANDLE hStdOut;*

*hStdOut = GetStdHandle(STD\_OUTPUT\_HANDLE);*

*if (hStdOut == INVALID\_HANDLE\_VALUE) {*

*return;*

*}*

*/\* set default gray on black attributes \*/*

*SetConsoleTextAttribute(hStdOut, FOREGROUND\_RED | FOREGROUND\_GREEN | FOREGROUND\_BLUE);*

*}*

*void con\_set\_pos(unsigned int x, unsigned int y)*

*{*

*COORD c;*

*HANDLE hStdOut;*

*hStdOut = GetStdHandle(STD\_OUTPUT\_HANDLE);*

*if (hStdOut == INVALID\_HANDLE\_VALUE) {*

*return;*

*}*

*c.X = (SHORT) x;*

*c.Y = (SHORT) y;*

*SetConsoleCursorPosition(GetStdHandle(STD\_OUTPUT\_HANDLE), c);*

*}*

*void con\_clear(void)*

*{*

*CONSOLE\_SCREEN\_BUFFER\_INFO csbi;*

*COORD c = { 0, 0 };*

*HANDLE hStdOut;*

*DWORD dwNumWritten;*

*hStdOut = GetStdHandle(STD\_OUTPUT\_HANDLE);*

*if (hStdOut == INVALID\_HANDLE\_VALUE) {*

*return;*

*}*

*if (!GetConsoleScreenBufferInfo(hStdOut, &csbi)) {*

*return;*

*}*

*/\* fill console buffer with spaces \*/*

*FillConsoleOutputCharacter(hStdOut, ' ', csbi.dwSize.X \* csbi.dwSize.Y, c, &dwNumWritten);*

*/\* set cursor to position 0,0 \*/*

*SetConsoleCursorPosition(hStdOut, c);*

*}*

*void fontSize(int a, int b){*

*HANDLE out = GetStdHandle(STD\_OUTPUT\_HANDLE);*

*PCONSOLE\_FONT\_INFOEX lpConsoleCurrentFontEx = new CONSOLE\_FONT\_INFOEX();*

*lpConsoleCurrentFontEx->cbSize = sizeof(CONSOLE\_FONT\_INFOEX);*

*GetCurrentConsoleFontEx(out, 0, lpConsoleCurrentFontEx);*

*lpConsoleCurrentFontEx->dwFontSize.X = a;*

*lpConsoleCurrentFontEx->dwFontSize.Y = b;*

*SetCurrentConsoleFontEx(out, 0, lpConsoleCurrentFontEx);*

*}*

*void setFill(unsigned int fg\_Color, unsigned int bg\_Color)*

*{*

*CONSOLE\_SCREEN\_BUFFER\_INFO csbi;*

*COORD c = { 0, 0 };*

*HANDLE hStdOut;*

*DWORD dwNumWritten;*

*hStdOut = GetStdHandle(STD\_OUTPUT\_HANDLE);*

*if (hStdOut == INVALID\_HANDLE\_VALUE) {*

*return;*

*}*

*if (!GetConsoleScreenBufferInfo(hStdOut, &csbi)) {*

*return;*

*}*

*/\* fill console buffer attributes \*/*

*FillConsoleOutputAttribute(hStdOut, (bg\_Color << 4) | fg\_Color, csbi.dwSize.X \* csbi.dwSize.Y, c, &dwNumWritten);*

*}*

*void animateTxt(char text[]){*

*for(int i;i<strlen(text);i++){*

*cout<<text[i];*

*\_sleep(50);*

*}*

*}*

*void fullscreen()*

*{*

*keybd\_event(VK\_MENU,0x38,0,0);*

*keybd\_event(VK\_RETURN,0x1c,0,0);*

*keybd\_event(VK\_RETURN,0x1c,KEYEVENTF\_KEYUP,0);*

*keybd\_event(VK\_MENU,0x38,KEYEVENTF\_KEYUP,0);*

*}*

*void gotoxy ( int x, int y )*

*{*

*COORD dwCursorPosition = { x, y };*

*SetConsoleCursorPosition ( GetStdHandle ( STD\_OUTPUT\_HANDLE ), dwCursorPosition );*

*}*

*void con\_wait\_key(void)*

*{*

*INPUT\_RECORD ir;*

*HANDLE hStdIn;*

*DWORD dwNumEvents, dwNumRead;*

*hStdIn = GetStdHandle(STD\_INPUT\_HANDLE);*

*if (hStdIn == INVALID\_HANDLE\_VALUE) {*

*return;*

*}*

*FlushConsoleInputBuffer(hStdIn);*

*/\* wait loop \*/*

*while (1) {*

*WaitForSingleObject(hStdIn, INFINITE);*

*if (GetNumberOfConsoleInputEvents(hStdIn, &dwNumEvents) && dwNumEvents) {*

*while (dwNumEvents--) {*

*if (ReadConsoleInput(hStdIn, &ir, 1, &dwNumRead) && dwNumRead) {*

*/\* if it's a key being released, return \*/*

*if (ir.EventType == KEY\_EVENT && ir.Event.KeyEvent.bKeyDown == 0) {*

*return;*

*}*

*}*

*}*

*}*

*}*

*}*

***// Segment 6 Contains main() Functions and the programmer runner.***

*#include "CarParkingControl.h"*

*#include<conio.h>*

*#define parkingSize 10*

*int main()*

*{*

*fullscreen();*

*fontSize(30,30);*

*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)*

*{*

*LoginPanel();*

*if (firstlog)*

*{*

*setColor(yellow, black);*

*cout << "NOTE: Your default \nUsername: 'iucrazz'\nPassword: 'iucrazz123' \nPlease change it for strong security reason..." << endl;*

*setColor(red,black);*

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

*}*

*setColor(green,black);*

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

*setColor(white,black);*

*cin >> option1;*

*switch (option1)*

*{*

*case '1':*

*{*

*enterAgain3:*

*string userNm;*

*string password;*

*setColor(green, black);*

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

*setColor(white,black);*

*cin>>userNm;*

*setColor(green,black);*

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

*password = InputPass();*

*if (checkLogin(userNm,password))*

*{*

*setColor(red,lightblue);*

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

*isLoginned = true;*

*goto DataBase2;*

*}*

*else*

*{*

*setColor(red,black);*

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

*isLoginned = false;*

*goto enterAgain3;*

*}*

*break;*

*}*

*case '2':*

*{*

*enterAgain4:*

*string tempPass;*

*setColor(green,black);*

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

*tempPass = InputPass();*

*if (DataBasePass==tempPass)*

*{*

*changePass:*

*setColor(green,black);*

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

*char ch3;*

*string newPass, confirmPass;*

*newPass = InputPass();*

*setColor(green,black);*

*cout<<"\n\nPlease enter ur Password again===\t";*

*confirmPass = InputPass();*

*if(newPass == confirmPass)*

*{*

*changePass(newPass); //Changin the Password*

*char choice;*

*string tempUser;*

*setColor(white,blue);*

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

*firstlog = false;*

*setColor(black,yellow);*

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

*setColor(white,black);*

*cin>>choice;*

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

*{*

*setColor(green,black);*

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

*setColor(white,black);*

*cin>>tempUser;*

*changeUsername(tempUser);*

*setColor(white,blue);*

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

*}*

*}*

*else*

*{*

*setColor(red,black);*

*cout<<"\n\nPassword not matching. Please try again!!";*

*goto changePass;*

*}*

*}*

*else*

*{*

*char cancel;*

*setColor(yellow,black);*

*cout << "\n\t\tWrong password!!! Please Try again... or enter 'x' to cancel..." << endl;*

*setColor(white,black);*

*cin>>cancel;*

*if(cancel == 'x' || cancel == 'X')*

*break;*

*else*

*goto enterAgain3;*

*}*

*break;*

*}*

*case '3':*

*char confirm1[2];*

*setColor(black,yellow);*

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

*setColor(white,black);*

*cin >> confirm1;*

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

*{*

*setColor(white,pink);*

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

*setColor(yellow,black);*

*goto last;*

*}*

*else*

*break;*

*default:*

*setColor(yellow,black);*

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

*break;*

*}*

*}*

*DataBase2:*

*while (isLoginned)*

*{*

*menu();*

*setColor(white,black);*

*cin >> option;*

*switch (option)*

*{*

*case '1':*

*{*

*if(carNo==parkingSize)*

*{*

*setColor(yellow,black);*

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

*break;*

*}*

*string Id;*

*bool ifexist=false;*

*setColor(green,black);*

*cout<<"\n\nEnter the carId:\t";*

*setColor(white,black);*

*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)*

*{*

*setColor(yellow,black);*

*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;*

*setColor(green,black);*

*cout << "\n\nEnter the CarId: ";*

*setColor(white,black);*

*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*

*{*

*setColor(yellow,black);*

*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;*

*setColor(green,black);*

*cout << "\n\nEnter the CarId: ";*

*setColor(white,black);*

*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)*

*{*

*setColor(yellow,black);*

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

*}*

*else if(recordFound==0)*

*{*

*setColor(yellow,black);*

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

*}*

*break;*

*}*

*case '4':*

*{*

*setColor(white,blue);*

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

*carNo=ParkingLotControl::getTotalCar();*

*setColor(black,yellow);*

*cout<<"\n\t\tTotal car Parked in the lot==";*

*setColor(green,black);*

*cout<<"\t"<<carNo<<" Cars"<<endl;*

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

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

*{*

*if(p[i].isLotEmpty)*

*{*

*setColor(white,green);*

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

*}*

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

*{*

*setColor(pink,black);*

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

*p[i].showTotalCarParked();*

*}*

*}*

*if(i==0)*

*{*

*setColor(yellow,black);*

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

*}*

*break;*

*}*

*case '5':*

*{*

*char confirm[1];*

*setColor(yellow,black);*

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

*setColor(white,black);*

*cin >> confirm;*

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

*{*

*setColor(white,pink);*

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

*goto loginPanel2;*

*}*

*else*

*break;*

*}*

*default:*

*setColor(yellow,black);*

*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!!!***