#include<iostream>

#include<conio.h>

#include<string>

#include <iomanip> // manipulator function

using namespace std;

class Driver

{

public:

string D\_name;

string D\_id;

int age;

string ph\_no;

bool li\_holder;

Driver(string na = " ", string id = " ", int a = 0, string ph = " ", bool lic = false)

{

}

void DisplayDriverInfo()

{

cout << right << "| " << D\_name;

cout << setw(12) << "| " << D\_id;

cout << setw(19) << "| " << age;

cout << setw(20) << "| " << ph\_no;

cout << setw(10) << "| ";

if (li\_holder == true)

{

cout << " YES " << endl;

}

else

{

cout << "NO " << endl;

}

}

};

class ListNode

{

public:

Driver data;

ListNode\* next;

ListNode(Driver d)

{

data = d;

}

};

class singly

{

ListNode\* head;

public:

singly()

{

head = NULL;

}

ListNode\* getHead()

{

return head;

}

void InsertAtStart(Driver d);

void DeleteAtStart();

void display();

};

void singly::InsertAtStart(Driver d)

{

ListNode\* newnode = new ListNode(d);

if (head == NULL)

{

head = newnode;

return;

}

else if (head != NULL)

{

newnode->next = head;

head = newnode;

return;

}

}

void singly::DeleteAtStart()

{

ListNode\* temp1 = head;

if (temp1 != NULL)

{

head = head->next;

delete temp1;

cout << "\n\n\t\t\t OPERATION PERFORMED SUCCESSFULLY!!!\n";

return;

}

else

{

cout << "\n\n\t\tDriver list is empty\n\n";

return;

}

}

void singly::display()

{

ListNode\* temp = head;

if (head == NULL)

{

cout << "List is Empty..." << endl;

return;

}

while (temp != NULL)

{

temp->data.DisplayDriverInfo();

temp = temp->next;

}

}

class Customer

{

public:

string name;

string contact;

int carNum;

Customer(string na = " ", string cont = " ", int cnum = 0)

{

}

void DisplayCustomerInfo()

{

cout << right << "| " << name;

cout << setw(12) << "| " << contact;

cout << setw(19) << "| " << carNum;

cout << setw(10) << "| ";

cout << endl;

}

};

class CNode

{

public:

Customer data;

CNode\* next;

CNode(Customer c)

{

data = c;

next = NULL;

}

};

class CQueue

{

CNode\* head;

public:

CQueue()

{

head = NULL;

}

void insertAtEnd(Customer c);

void display();

};

void CQueue::insertAtEnd(Customer c)

{

CNode\* newnode = new CNode(c);

CNode\* temp = head;

if (temp == NULL)

{

head = newnode;

return ;

}

else

{

while (temp->next != NULL) {

temp = temp->next;

}

if (temp->next == NULL) {

temp->next = newnode;

return ;

}

else

{

return ;

}

}

}

void CQueue::display()

{

CNode\* temp = head;

if (head == NULL)

{

cout << "List is Empty..." << endl;

return;

}

while (temp != NULL)

{

temp->data.DisplayCustomerInfo();

temp = temp->next;

}

}

class Driver\_List

{

public:

singly D\_List;

Driver\_List()

{

}

void Push(Driver d);

Driver Pop();

Driver Top();

};

void Driver\_List::Push(Driver d)

{

D\_List.InsertAtStart(d);

}

Driver Driver\_List::Pop()

{

Driver Dr;

if (D\_List.getHead()==NULL)

{

cout << "\n\n\t\tDriver list is already empty!!!\n\n";

return Dr;

}

Dr = D\_List.getHead()->data;

D\_List.DeleteAtStart();

return Dr;

}

Driver Driver\_List::Top()

{

Driver Dr;

if (D\_List.getHead() == NULL)

{

return Dr;

}

Dr = D\_List.getHead()->data;

return Dr;

}

class Car

{

public:

int car\_no;

string car\_name;

string engine\_no;

int engine\_cc;

long int car\_price;

Car(int cn = 0, string name = " ", string en = " ", int ecc = 0,long int pric=0)

{

car\_no = cn;

car\_name = name;

engine\_no = en;

engine\_cc = ecc;

car\_price = pric;

}

void DisplayCarInfo()

{

cout << right << "| " << car\_no;

cout << setw(19) << "| "<<car\_name;

cout << setw(19) << "| "<<engine\_no;

cout << setw(20) << "| "<<engine\_cc;

cout << setw(10) << "| "<<car\_price<<endl<<endl;

}

};

class Node

{

public:

Car data;

Node\* left = NULL;

Node\* right = NULL;

Node(Car temp)

{

data = temp;

left = NULL;

right = NULL;

}

};

class manage\_cars

{

Node\* root;

public:

manage\_cars()

{

root = NULL;

}

Node\* get\_root()

{

return root;

}

void insertion(Car);

void search(int);

void dispaly\_preorder(Node\*);

void deletetion(int val);

bool car\_exist(int);

};

void manage\_cars::deletetion(int val)

{

Node\* temp = root;

if (root == NULL)

{

cout << "\n\n\t\tList is already empty " << endl << endl;

}

else

{

Node\* parent = root;

while (temp != NULL)

{

if (val == temp->data.car\_no)

{

break;

}

else if (val < temp->data.car\_no)

{

parent = temp;

temp = temp->left;

}

else

{

parent = temp;

temp = temp->right;

}

if (temp == NULL)

{

cout << "\n\n\t\tYOUR CHOICE IS INCORRECT!!! \n" << endl;

return;

}

}

if (temp->left == NULL && temp->right == NULL) //for leaf

{

if (temp == root)

{

root = NULL;

}

else if (parent->left == temp)

{

parent->left = NULL;

}

else

{

parent->right = NULL;

}

delete temp;

cout << "\n\n\t\t\tTHANKS FOR SELECTING THIS CAR\n\n";

}

else if (temp->left == NULL || temp->right == NULL) //for single child

{

if (temp->left == NULL)

{

if (temp == root)

{

root = temp->right;

}

else if (parent->left == temp)

{

parent->left = temp->right;

}

else

{

parent->right = temp->right;

}

delete temp;

}

else

{

if (temp == root)

{

root = temp->left;

}

else if (parent->left == temp)

{

parent->left = temp->left;

}

else

{

parent->right = temp->left;

}

delete temp;

}

cout << "\n\n\t\t\tTHANKS FOR SELECTING THIS CAR\n\n";

}

else // for two childs

{

Node\* temp2 = temp->right;

Node\* p = NULL;

while (temp2->left != NULL) //inorder successor

{

p = temp2;

temp2 = temp2->left;

}

temp->data = temp2->data;

if (temp2->left == NULL && temp2->right == NULL) //for leaf

{

if (p->left == temp2)

{

p->left = NULL;

}

else

{

p->right = NULL;

}

delete temp2;

cout << "\n\n\t\t\tTHANKS FOR SELECTING THIS CAR\n\n";

}

else if (temp2->left == NULL || temp2->right == NULL) //for single child

{

if (temp2->left == NULL)

{

if (p->left == temp2)

{

p->left = temp2->right;

}

else

{

p->right = temp2->right;

}

delete temp2;

}

cout << "\n\n\t\t\tTHANKS FOR SELECTING THIS CAR\n\n";

}

}

}

}

void manage\_cars::insertion(Car c)

{

Node\* newnode = new Node(c);

int value = c.car\_no;

if (root == NULL)

{

root = newnode;

return;

}

else

{

Node\* temp = root;

while (temp != NULL)

{

if (value > temp->data.car\_no)

{

if (temp->right == NULL)

{

temp->right = newnode;

return;

}

temp = temp->right;

}

else

{

if (temp->left == NULL)

{

temp->left = newnode;

return;

}

temp = temp->left;

}

}

}

}

void manage\_cars::search(int val)

{

Node\* temp = root;

if (root == NULL)

{

cout << "\n\n\t\tList is empty " << endl << endl;

return;

}

else

{

while (temp != NULL)

{

if (val == temp->data.car\_no)

{

temp->data.DisplayCarInfo();

return;

}

else if (val < temp->data.car\_no)

{

temp = temp->left;

}

else

{

temp = temp->right;

}

if (temp == NULL)

{

cout << "\n\n\t\t\tVALUE NOT FFOUND!!! \n" << endl;

return;

}

}

}

}

bool manage\_cars::car\_exist(int val)

{

Node\* temp = root;

if (root == NULL)

{

return false;

}

else

{

while (temp != NULL)

{

if (val == temp->data.car\_no)

{

return true;

}

else if (val < temp->data.car\_no)

{

temp = temp->left;

}

else

{

temp = temp->right;

}

if (temp == NULL)

{

return false;

}

}

}

}

void manage\_cars::dispaly\_preorder(Node\* temp) // PLR

{

if (root == NULL)

{

cout << "\n\n\t\tLIST IS EMPTY!!! " << endl << endl;

}

else

{

if (temp == NULL)

{

return;

}

temp->data.DisplayCarInfo();

dispaly\_preorder(temp->left);

dispaly\_preorder(temp->right);

}

}

class Admin

{

public:

//int total\_sales;

manage\_cars cars;

Driver\_List drivers;

CQueue custo;

Admin()

{

}

void add\_car(Car c)

{

cars.insertion(c);

}

void view\_cars()

{

cout << endl << endl;

cout << left << setw(20) << "Car # " << "|" << setw(20) << "Name" << "|"

<< setw(20) << "Engine #" << "|" << setw(15) << " CC " << "|" << setw(20) << "Price" << endl;

cars.dispaly\_preorder(cars.get\_root());

}

void search\_car(int n)

{

cars.search(n);

}

void add\_driver(Driver d)

{

drivers.Push(d);

}

void remove\_driver()

{

drivers.Pop();

}

void view\_driver\_list()

{

cout << endl << endl;

cout << left << setw(20) << "|Name " << "|" << setw(20) << "ID #" << "|"

<< setw(20) << "Age" << "|" << setw(15) << "Phone # " << "|" << setw(20) << "License Holder" << endl;

drivers.D\_List.display();

}

void sale\_car(Car c)

{

cars.insertion(c);

}

void rent\_car(int a)

{

cars.deletetion(a);

}

void lease\_car(int a)

{

cars.deletetion(a);

}

void buy\_car(int a)

{

cars.deletetion(a);

}

void summary\_report()

{

cout << endl << endl;

cout << left << setw(20) << "|Name " << "|" << setw(20) << "Phone #" << "|"

<< setw(20) << "Car #" << "|" << endl;

custo.display();

}

};

void menu()

{

manage\_cars k;

Driver d;

Car c1;

Admin a;

Customer c2;

L1:

system("cls");

cout << "\n\n" << "\t\t\t\t==========================================================";

cout << "\n\n" <<"\t\t\t\t =================================================";

cout << "\n\n" << "\t\t\t\t\t ~ CARS MANAGEMENT SYSTEM ~";

cout << "\n\n" << "\t\t\t\t =================================================";

cout << "\n\n" << "\t\t\t\t==========================================================";

cout << "\n\n\n\n";

int ch;

cout << "\t\t\t\t\t1.MANAGE CARS\n";

cout << "\t\t\t\t\t2.MANAGE DRIVERS\n";

cout << "\t\t\t\t\t3.DEALING CARS\n";

cout << "\t\t\t\t\t4.CUSTOMERS SUMMARY REPORT\n";

cout << "\t\t\t\t\t5.EXIT.\n";

cout << "\n\tPlease enter a number : ";

cin >> ch;

switch (ch)

{

case 1:

{

L2:

system("cls");

cout << "\n\n" << "\t\t\t\t\t\t\t=============================";

cout << "\n\n" << "\t\t\t\t\t\t\t======== MANAGE CARS ========";

cout << "\n\n" << "\t\t\t\t\t\t\t=============================";

int ch1;

cout << "\n\t\t\t\t\t0.DEFAULT CARS\n";

cout << "\t\t\t\t\t1.VIEW CARS\n";

cout << "\t\t\t\t\t2.ADD CARS\n";

cout << "\t\t\t\t\t3.SEARCH CARS\n";

cout << "\t\t\t\t\t4.BACK TO MENU\n";

cout << "\t\t\t\t\t5.EXIT\n";

cout << "\n\tPlease enter a number : ";

cin >> ch1;

switch (ch1)

{

case 0:

c1.car\_no = 1;

c1.engine\_no = 56;

c1.engine\_cc = 1600;

c1.car\_name = "suzuki";

c1.car\_price = 360000;

a.cars.insertion(c1);

c1.car\_no = 2;

c1.engine\_no = 56;

c1.engine\_cc = 1800;

c1.car\_name = "civic";

c1.car\_price = 360000;

a.cars.insertion(c1);

break;

case 1:

a.view\_cars();

break;

case 2:

cout << "\nEnter car no :";

cin >> c1.car\_no;

while (a.cars.car\_exist(c1.car\_no))

{

cout << "\n\n\t\tThis car number is already exist!!!\n";

cout << "\nPlease enter again car no :";

cin >> c1.car\_no;

}

cout << "\nEnter engine no : ";

cin >> c1.engine\_no;

cout << "\nEnter engine cc : ";

cin >> c1.engine\_cc;

c1.engine\_cc = abs(c1.engine\_cc);

cin.ignore();

cout << "\nEnter car name : ";

getline(cin, c1.car\_name);

cout << "\nEnter car price :";

cin >> c1.car\_price;

a.add\_car(c1);

break;

case 3:

int num;

cout << "\nEnter car no :";

cin >> num;

a.search\_car(num);

break;

case 4:

goto L1;

break;

case 5:

exit(0);

default:

cout << "\n\n\t\tYou entered an invalid number!!!\n\n";

}

system("pause");

goto L2;

break;

}

case 2:

{

L3:

system("cls");

cout << "\n\n" << "\t\t\t\t\t\t\t=============================";

cout << "\n\n" << "\t\t\t\t\t\t\t======== MANAGE DRIVERS =====";

cout << "\n\n" << "\t\t\t\t\t\t\t=============================";

int ch2;

cout << "\n\t\t\t\t\t0.Deault DRIVER\n";

cout << "\t\t\t\t\t1.ADD DRIVER\n";

cout << "\t\t\t\t\t2.REMOVE DRIVER\n";

cout << "\t\t\t\t\t3.VIEW DRIVER LIST\n";

cout << "\t\t\t\t\t4.BACK TO MENU\n";

cout << "\t\t\t\t\t5.EXIT\n";

cout << "\n\tPlease enter a number : ";

cin >> ch2;

switch (ch2)

{

case 0:

d.D\_name = "junaid";

d.D\_id = "23444";

d.ph\_no = 0232353;

d.age = 21;

d.li\_holder = true;

a.add\_driver(d);

d.D\_name = "zaryab";

d.D\_id = "234565";

d.ph\_no = 02353453;

d.age = 18;

d.li\_holder = false;

a.add\_driver(d);

break;

case 1:

char c;

cin.ignore();

cout << "\nEnter you name :";

getline(cin, d.D\_name);

cout << "\nEnter you ID number :";

getline(cin, d.D\_id);

cout << "\nEnter you contact mumber :";

getline(cin, d.ph\_no);

cout << "\nEnter you age :";

cin >> d.age;

d.age = abs(d.age);

cout << "\nAre you license holder? (Y/N) :";

cin >> c;

while ((c != 'Y') && (c != 'y') && (c != 'n') && (c != 'N'))

{

cout << "Invalid" << endl;

cout << "\nAre you license holder? (Y/N) :";

cin >> c;

}

if (c == 'Y' || c == 'y')

{

d.li\_holder = true;

}

else

{

d.li\_holder = false;

}

a.add\_driver(d);

break;

case 2:

a.remove\_driver();

break;

case 3:

a.view\_driver\_list();

break;

case 4:

goto L1;

break;

case 5:

exit(0);

default:

cout << "\n\t\tYou entered an invalid number!!!\n\n";

}

system("pause");

goto L3;

break;

}

case 3:

{

L4:

system("cls");

cout << "\n\n" << "\t\t\t\t\t\t\t=============================";

cout << "\n\n" << "\t\t\t\t\t\t\t======== DEALING CARS =======";

cout << "\n\n" << "\t\t\t\t\t\t\t=============================";

int ch3;

cout << "\n\t\t\t\t\t1.SALE CARS\n";

cout << "\t\t\t\t\t2.RENT CAR\n";

cout << "\t\t\t\t\t3.LEASE CAR\n";

cout << "\t\t\t\t\t4.BUY CAR\n";

cout << "\t\t\t\t\t5.BACK TO MENU\n";

cout << "\t\t\t\t\t6.EXIT\n";

cout << "\n\tPlease enter a number : ";

cin >> ch3;

switch (ch3)

{

case 1:

cout << "\nEnter car no :";

cin >> c1.car\_no;

while (a.cars.car\_exist(c1.car\_no))

{

cout << "\n\t\tThis car number is already exist!!!\n";

cout << "\nPlease enter again car no :";

cin >> c1.car\_no;

}

cout << "\nEnter engine no : ";

cin >> c1.engine\_no;

cout << "\nEnter engine cc : ";

cin >> c1.engine\_cc;

cin.ignore();

cout << "\nEnter car name : ";

getline(cin, c1.car\_name);

cout << "\nEnter car price :";

cin >> c1.car\_price;

cin.ignore();

//customer data

cout << "\nEnter your name : ";

getline(cin, c2.name);

cout << "\nEnter your contact : ";

getline(cin, c2.contact);

c2.carNum = c1.car\_no;

a.custo.insertAtEnd(c2);

a.sale\_car(c1);

break;

case 2:

a.view\_cars();

if (a.cars.get\_root() != NULL)

{

//customer data

cin.ignore();

cout << "\nEnter your name : ";

getline(cin, c2.name);

cout << "\nEnter your contact : ";

getline(cin, c2.contact);

char ch;

bool dr;

cout << "\nDo you need a driver (Y/N): ";

cin >> ch;

while ((ch != 'Y') && (ch != 'y') && (ch != 'n') && (ch != 'N'))

{

cout << "Invalid" << endl;

cout << "\nDo you need a driver (Y/N) :";

cin >> ch;

}

if (ch == 'Y' || ch == 'y')

{

dr = true;

}

else

{

dr = false;

}

if (dr == true)

{

a.remove\_driver();

}

cout << "'\n\t\t\t\t OUR PER DAY CHARGES OF RENT ARE 3000 ruppies";

cout << "'\n\t\t\t\t OUR PER DAY CHARGES OF DRIVER ARE 2000 ruppies\n";

int pricea, days;

cout << endl;

cout << "\n\t FOR HOW MANY DAYS YOU WANT TO RENT A CAR. : ";

cin >> days;

g1:

cout << "\n\tEnter a car no which do you want : ";

cin >> c1.car\_no;

if (a.cars.car\_exist(c1.car\_no))

{

a.rent\_car(c1.car\_no);

}

else

{

cout << "\n\t\tYou entered an invalid number!!!\n\n";

goto g1;

}

c2.carNum = c1.car\_no;

a.custo.insertAtEnd(c2);

if (dr == true)

{

pricea = (days \* 3000) + (days\*2000);

}

else

pricea = days \* 3000;

cout << "\n\t\t YOUR RENT CHARGES FOR " << days << " DAYS ARE :" << pricea << endl;

}

break;

case 3:

a.view\_cars();

if (a.cars.get\_root() != NULL)

{

//customer data

cin.ignore();

cout << "\nEnter your name : ";

getline(cin, c2.name);

cout << "\nEnter your contact : ";

getline(cin, c2.contact);

g2:

cout << "\nEnter a car no which do you want : ";

cin >> c1.car\_no;

if (a.cars.car\_exist(c1.car\_no))

{

a.lease\_car(c1.car\_no);

}

else

{

cout << "\n\t\tYou entered an invalid number!!!";

goto g2;

}

c2.carNum = c1.car\_no;

a.custo.insertAtEnd(c2);

cout << "\n\t WE CHARGE 5% EXTRA TAX FOR 1 YEAR & 10% EXTRA FOR 2 YEAES ";

int year;

long int priceb;

cout << "\n\t\t FOR HOW MANY YEARS YOU WANT INSTALLMENT : ";

cin >> year;

while (year != 1 && year != 2)

{

cout << "\n\t\tYOU ENTERED AN INVALID YEAR!!!\n";

cout << "\n\t\t PLEASE ENTER AGAIN FOR HOW MANY YEARS YOU WANT INSTALLMENT : ";

cin >> year;

}

cout << "ORIGNAL PRICE OF YOUR SELECTED CAR IS : " << c1.car\_price << endl;

if (year == 1)

{

priceb = (c1.car\_price \* 5);

priceb = priceb / 100;

priceb = c1.car\_price + priceb;

}

else if (year == 2)

{

priceb = (c1.car\_price \* 10);

priceb = priceb / 100;

priceb = c1.car\_price + priceb;

}

cout << " \n\t\tYOUR TOTAL PAYMENT IS :" << priceb;

cout << endl;

cout << " \n\tYOUR PER MONTH INSTALLMENT IS :" << priceb / 12 << endl;

}

break;

case 4:

a.view\_cars();

if (a.cars.get\_root() != NULL)

{

//customer data

cin.ignore();

cout << "\nEnter your name : ";

getline(cin, c2.name);

cout << "\nEnter your contact : ";

getline(cin, c2.contact);

g3:

cout << "\n\tEnter a car no which do you want : ";

cin >> c1.car\_no;

if (a.cars.car\_exist(c1.car\_no))

{

a.buy\_car(c1.car\_no);

}

else

{

cout << "\n\t\tYou entered an invalid number!!!\n\n";

goto g3;

}

c2.carNum = c1.car\_no;

a.custo.insertAtEnd(c2);

cout << " \n\t\tYOUR TOTAL PAYMENT IS :" << c1.car\_price << endl;

cout << endl;

}

break;

case 5:

goto L1;

break;

case 6:

exit(0);

default:

cout << "\n\t\tYou entered an invalid number!!!\n\n";

}

system("pause");

goto L4;

break;

}

case 4:

system("cls");

cout << "\n\n" << "\t\t\t\t\t\t\t=============================";

cout << "\n\n" << "\t\t\t\t\t\t\t======== SUMMARY REPORT =====";

cout << "\n\n" << "\t\t\t\t\t\t\t=============================";

cout << endl;

a.summary\_report();

break;

case 5:

exit(0);

break;

default:

cout << "\n\n\t\t\tYou entered an invalid number!!!!\n\n\n";

}

system("pause");

goto L1;

}

int main()

{

menu();

}