DoublyLinkList-(Showroom)

**CODE:**

#include<iostream>

#include<string>

#include<windows.h>

using namespace std;

HANDLE h = GetStdHandle(STD\_OUTPUT\_HANDLE);

class car

{

public:

    string company;

    string mname;

    string myear;

    int pprice;

    int sprice;

    string number;

    car()

    {

        company = "Nocompany";

        mname = "NoModelName";

        myear = "Nomyear";

        pprice = -1;

        sprice = -1;

        number = "";

    }

    void input()

    {

        cout << " Company        : ";

        cin >> company;

        cout << " Model Name     : ";

        cin >> mname;

        cout << " Model Year     : ";

        cin >> myear;

        cout << " Number         : ";

        cin >> number;

        cout << " Purchase Price : ";

        cin >> pprice;

    }

    void displaydetails()

    {

        SetConsoleTextAttribute(h, 11);

        cout << "\n\t\t\t Car Details \n";

        SetConsoleTextAttribute(h,10);

        cout << "\n Company         : " << company << endl;

        cout << "\n Model NAME      : " << mname << endl;

        cout << "\n Model YEAR      : " << myear << endl;

        cout << "\n Purchased PRICE : " << pprice << endl;

        cout << "\n Number          : " << number << endl;

        SetConsoleTextAttribute(h, 7);

    }

    string getnumber()

    {

        return number;

    }

};

struct node

{

    car data;

    node\* next;

    node\* prev;

    node()

    {

        next = prev = NULL;

    }

};

class doublyLinkList

{

private:

    node\* head;

    node\* soldhead;

public:

    car c;

    int soldcount = 0;

    doublyLinkList()

    {

        head = NULL;

        soldhead = NULL;

    }

    bool isempty()

    {

        if (head == NULL)

            return true;

        else return false;

    }

    node\* getnode(car c)

    {

        node\* n = new node;

        n->data = c;

        n->next = NULL;

        n->prev = NULL;

        return n;

    }

    void insertatfront(car c)

    {

        node\* n = getnode(c);

        if (isempty())

        {

            head = n;

        }

        else

        {

            n->next = head;

            head->prev = n;

            head = n;

        }

    }

    void insertatend(car c)

    {

        node\* n = getnode(c);

        node\* temp = head;

        while (temp->next != NULL)

        {

            temp = temp->next;

        }

        temp->next = n;

        n->prev = temp;

    }

    void insertinsold(node\* temp)

    {

        node\* n = new node;

        n->data = temp->data;

        if (soldhead == NULL)        //newlist

        {

            soldhead = n;

            return;

        }

        else

        {

            n->next = soldhead;

            soldhead->prev = n;

            soldhead = n;

        }

    }

    void insert()

    {

        car c;

        int ch, repeat = 1;

        while (repeat == 1)

        {

            system("cls");

            SetConsoleTextAttribute(h, 11);

            cout << "\n\n\t\t\t\t INSERTION-(Dlinklist)\n";

            cout << " 1. Insert at Front \n";

            cout << " 2. Insert at End \n";

            cout << " 0. Main Menu\n";

            SetConsoleTextAttribute(h, 7);

            cin >> ch;

            switch (ch)

            {

            case 1:

                system("cls");

                c.input();

                insertatfront(c);

                break;

            case 2:

                system("cls");

                c.input();

                insertatend(c);

                break;

            case 0:

                repeat++;

                system("cls");

                return;

                break;

            default:

                SetConsoleTextAttribute(h, 4);

                cout << " \n Invalid Input ! Choose from the given Options !!!\n";

                SetConsoleTextAttribute(h, 7);

            }

        }

    }

    void deletefromfront()

    {

        if (isempty())

        {

            SetConsoleTextAttribute(h, 4);

            cout << "\n The List is Empty, Nothing to Delete !!!\n";

            SetConsoleTextAttribute(h, 7);

        }

        else

        {

            node\* temp = head;

            head = head->next;

            SetConsoleTextAttribute(h, 4);

            cout << " \n Car with Number ' " << temp->data.getnumber() << " ' DELETED !!!";

            SetConsoleTextAttribute(h, 7);

            //copy to New list

            /////////////////////////////////////////////////////////////////

            insertinsold(temp);

            //////////////////////////////////////////////////////////////////

            delete temp;

        }

    }

    void deletefromend()

    {

        if (isempty())

        {

            SetConsoleTextAttribute(h, 4);

            cout << "\n The List is Empty, Nothing to Delete !!!\n";

            SetConsoleTextAttribute(h, 7);

        }

        else

        {

            node\* temp = head;

            while (temp->next != NULL)

            {

                temp = temp->next;

            }

            SetConsoleTextAttribute(h, 4);

            cout << " \n Car with Number ' " << temp->data.getnumber() << " ' DELETED !!!";

            SetConsoleTextAttribute(h, 7);

            //copy to New list

            //////////////////////////////////////////////////////////////////////

            insertinsold(temp);

            //////////////////////////////////////////////////////////////////////

            temp->prev->next = NULL;

            delete temp;

        }

    }

    void deletehavingnumber(string num)

    {

        if (isempty())

        {

            SetConsoleTextAttribute(h, 4);

            cout << "\n The List is Empty, Nothing to Delete !!!\n";

            SetConsoleTextAttribute(h, 7);

        }

        else

        {

            if (head->data.getnumber() == num)         //if its the first node

            {

                deletefromfront();

            }

            else

            {

                node\* temp = head;

                while (temp->next != NULL)

                {

                    temp = temp->next;

                    if (temp->data.getnumber() == num)

                    {

                        if (temp->next == NULL)      //if last node then mempry access violation error

                        {

                            deletefromend();

                            return;

                        }

                        else                         //if the node in inbetwn

                        {

                            node\* t = temp;

                            t->next->prev = t->prev;

                            t->prev->next = t->next;

                            SetConsoleTextAttribute(h, 4);

                            cout << " \n Car with Number ' " << temp->data.getnumber() << " ' DELETED !!!";

                            SetConsoleTextAttribute(h, 7);

                            insertinsold(temp);

                            delete t;

                            return;

                        }

                    }

                }

            }

        }

    }

    void deletion()

    {

        int ch;

        soldcount++;

        string num;

        SetConsoleTextAttribute(h, 11);

        cout << "\n\n\t\t\t\t Deletion-(Dlinklist)\n";

        cout << " 1. Delete at Front \n";

        cout << " 2. Delete at End \n";

        cout << " 3. Delete Having Number\n";

        SetConsoleTextAttribute(h, 7);

        cin >> ch;

        switch (ch)

        {

        case 1:

            deletefromfront();

            break;

        case 2:

            deletefromend();

            break;

        case 3:

            cout << " Enter number of the CAR : ";

            cin >> num;

            deletehavingnumber(num);

            break;

        default:

            SetConsoleTextAttribute(h, 4);

            cout << " \n Invalid Input ! Choose from the given Options !!!\n";

            SetConsoleTextAttribute(h, 7);

        }

    }

    void display\_list()

    {

        node\* temp = head;

        if (isempty())

        {

            SetConsoleTextAttribute(h, 4);

            cout << "\n The List is Empty, No CARS to SHOW!!\n";

            SetConsoleTextAttribute(h, 7);

        }

        else

        {

            while (temp != NULL)

            {

                temp->data.displaydetails();

                cout << endl;

                temp = temp->next;

            }

        }

    }

    void display\_sold()

    {

        node\* temp = soldhead;

        if (isempty())

        {

            SetConsoleTextAttribute(h, 4);

            cout << "\n The  SOLD List is Empty, No CARS Sold Yet!!\n";

            SetConsoleTextAttribute(h, 7);

        }

        else

        {

            while (temp != NULL)

            {

                temp->data.displaydetails();

                cout << endl;

                temp = temp->next;

            }

        }

    }

    void searchwithnumber()

    {

        int v=0;

        string num;

        cout << " Enter Number of CAR : ";

        cin >> num;

        if (isempty())

        {

            SetConsoleTextAttribute(h, 4);

            cout << "\n The List is Empty!!!\n";

            SetConsoleTextAttribute(h, 7);

        }

        else

        {

            node\* temp = head;

            while (temp != NULL)

            {

                if (temp->data.getnumber() == num)

                {

                    v++;

                    SetConsoleTextAttribute(h, 11);

                    cout << " \n\n\t\t CAR FOUND in Inventory !!!  \n";

                    SetConsoleTextAttribute(h, 7);

                    temp->data.displaydetails();

                    return ;

                }

                temp = temp->next;

            }

            if (v == 0)

            {

                cout << " \n\n\t\t CAR doesnot exist !!! \n";

                return ;

            }

        }

    }

    /////////////////////////////////////////////////

    int gettotalsold()

    {

        return soldcount;

    }

    void soldornot()

    {

        int ch;

        string num;

        SetConsoleTextAttribute(h, 13);

        cout << " Enter Number of CAR : ";

        cin >> num;

        SetConsoleTextAttribute(h, 7);

        if (isempty())

        {

            SetConsoleTextAttribute(h, 4);

            cout << "\n The List is Empty !!!\n";

            SetConsoleTextAttribute(h, 7);

        }

        else

        {

            node\* temp = head;

            node\* temp2 = soldhead;

            while (temp != NULL)

            {

                if (temp->data.getnumber() == num)

                {

                    SetConsoleTextAttribute(h, 11);

                    cout << " \n\n\t\t CAR FOUND in Inventory !!!  \n";

                    cout << " \n\t\t\t NOT SOLD yet!!!!";

                    SetConsoleTextAttribute(h, 13);

                    cout << "\n\n\t\t Press 1 to see details of car \n";

                    cout << "\t\t Press 0 to goto Main Menu\n";

                    SetConsoleTextAttribute(h, 7);

                    cin >> ch;

                    switch (ch)

                    {

                    case 1:

                        system("cls");

                        cout << " \n\t\t\t NOT SOLD yet!!!!\n\n";

                        temp->data.displaydetails();

                        break;

                    case 0:

                        return;

                        break;

                    default:

                        SetConsoleTextAttribute(h, 4);

                        cout << " \n Invalid Input ! Choose from the given Options !!!\n";

                        SetConsoleTextAttribute(h, 7);

                    }

                    return;

                }

                temp = temp->next;

            }

            while (temp2 != NULL)

            {

                if (temp2->data.getnumber() == num)

                {

                    SetConsoleTextAttribute(h, 11);

                    cout << " \n\n\t\t CAR FOUND in SOLD LIST !!!  \n\n";

                    cout << " \n\t\t\t SOLD!!!";

                    SetConsoleTextAttribute(h, 13);

                    cout << "\n\n\t\t Press 1 to see details of car \n";

                    cout << "\t\t Press 0 to goto Main Menu\n";

                    SetConsoleTextAttribute(h, 7);

                    cin >> ch;

                    switch (ch)

                    {

                    case 1:

                        system("cls");

                        cout << " \n\t\t\t SOLD!!!\n\n";

                        temp2->data.displaydetails();

                        break;

                    case 0:

                        return;

                        break;

                    default:

                        SetConsoleTextAttribute(h, 4);

                        cout << " \n Invalid Input ! Choose from the given Options !!!\n";

                        SetConsoleTextAttribute(h, 7);

                    }

                    return;

                }

                temp2 = temp2->next;

            }

        }

    }

    void details()

    {

        int ch;

        SetConsoleTextAttribute(h, 11);

        cout << "\n\n\t\t\t Details of Sales \n";

        cout << " 1. Total Number of cars Sold \n";

        cout << " 2. Details of Cars Sold\n";

        cout << " 3. Check the car is sold or not!\n";

        SetConsoleTextAttribute(h, 7);

        cin >> ch;

        switch (ch)

        {

        case 1:

            system("cls");

            SetConsoleTextAttribute(h, 11);

            cout << " Total Cars Sold : " << gettotalsold() << endl;

            SetConsoleTextAttribute(h, 7);

            break;

        case 2:

            system("cls");

            display\_sold();

            break;

        case 3:

            system("cls");

            soldornot();

            break;

        default:

            SetConsoleTextAttribute(h, 4);

            cout << " \n Invalid Input ! Choose from the given Options !!!\n";

            SetConsoleTextAttribute(h, 7);

        }

    }

};

int main()

{

    doublyLinkList d;

    car c;

    int repeat = 1, ch = -1,chh=-1;

    string num;

    while (repeat == 1)

    {

        system("cls");

        SetConsoleTextAttribute(h, 11);

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

        cout << " \n\t\t\t MENU \t\t\n";

        cout << "\n 1. Insert New Car ";

        cout << "\n 2. Delete Car ";

        cout << "\n 3. Search Car ";

        cout << "\n 4. Details of Sale ";

        cout << "\n 4. Display ALL Car Details   \n";

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

        SetConsoleTextAttribute(h, 7);

        cin >> ch;

        switch (ch)

        {

        case 1:

            system("cls");

            d.insert();

            break;

        case 2:

            system("cls");

            d.deletion();

            cout << " \n 0. Main Menu\n";

            cin >> chh;

            if (chh == 0)

            {

                break;

            }

            break;

        case 3:

            system("cls");

            d.searchwithnumber();

            break;

        case 4:

            system("cls");

            d.details();

            break;

        case 5:

            system("cls");

            d.display\_list();

            cout << " \n 0. Main Menu\n";

            cin >> chh;

            if (chh == 0)

            {

                break;

            }

            break;

        default:

            cout << " \n Invalid Input! Choose from the given options !!! \n";

        }

    }

    system("pause");

    return 0;

}

**OUTPUT:**

