

Object Oriented Programming Final Project Train Management System



Written by:
Umair Ishtiaq
Reg no:
BCS203191
Section:
S2

CODE

```
#include<windows.h>
```

```
#include<string>
```

```
#include<conio.h>
```

```
using namespace std;
```

 $\{$

```
void WelcomeMessage()
```

 $\{$

```
SetConsoleTextAttribute(colour, 3);
```

$$\frac{1}{\sqrt{1-\beta^2}}$$

```
cout << "\t\t\t\t\t\n";
```

```
cout << "\t\t\t| | _ | / ` / \ \ ||V|| _ | / \ \ | _ ) \ \ | | | | \ \ W /
```

```
cout<< "\t\t|\/\| |__|__\\_/_/||_| |\_/_ | \/_~\| |__|\/\|  
/_~\| | \n";
```

```
cout << "\t\t\t\t\t\n";
```

[illegible]

```

        cout << "\t\t\t | $$ | $$| $$ | $$ | $$ /   $$| $$| $$  \\\$          \n";
        cout << "\t\t\t | $$__/$$| $$ | $$ | $$| $$$$$$| $$| $$          \n";
        cout << "\t\t\t  \\\$ $ $| $$ | $$ | $$ \\\$  $ $| $$| $$          \n";
        cout << "\t\t\t  \$$$$$ \\\$ \\\$ \\\$ \\\$ \$$$$$ \\\$ \\\$          \n";
        cout << "\n\n\n\t\t\t  _____  _  _  _          \n";
        cout << "\t\t\t |   \\\$   | \\\$   | \\\$   | \\\$          \n";
        cout << "\t\t\t  \$$$$$ _____ | $$_____ | $$_  \\\$ _____  \n";
        cout << "\t\t\t | $$ /   \\\$ \\\$ \\\$ \\\$ \\\$ | \\\$ |   \\\$ /   \\\$   \n";
        cout << "\t\t\t | $$ | $$$$$$| $$$$$$\$$$$$ | $$ \$$$$$\$ \\\$ | $$$$$$\$ \\\$
\n";

        cout << "\t\t\t | $$ \\\$ \\\$ \\\$ | $$ | $$ | $$ _ | $$ /   $$| $$ | $$   \n";
        cout << "\t\t\t _ | $$ _ _\$$$$$\$ \\\$ | $$ | $$ | \\\$ | \\\$ \\\$ | $$ _ | $$   \n";
        cout << "\t\t\t | $$ \\\$   $$| $$ | $$ \\\$ $ $| $$ \\\$  $ $ \\\$  $ $   \n";
        cout << "\t\t\t  \$$$$$ \$$$$$ \\\$ \\\$ \\\$ \\\$ \\\$ \$$$$$ \$$$$$
\n";

        cout << "\t\t\t                               | $$   \n";
        cout << "\t\t\t                               | $$   \n";
        cout << "\t\t\t                               \\\$   \n";
    }

    virtual ~Discription()
    {
        cout << "\n*****Studentname destructor called*****\n";
    }
};

class Passenger :public Discription
{
protected:
    string NameOfPassenger;
    string PassengerGender;

```

```

public:
    void InputName()
    {
        HANDLE colour = GetStdHandle(STD_OUTPUT_HANDLE);
        SetConsoleTextAttribute(colour, 11);
        cout << "Enter your Name : ";
        getline(cin, NameOfPassenger);
    }
    void InputGender()
    {
        cout << "Enter your Gender : ";
        getline(cin, PassengerGender);
    }
    void InputPassenger()
    {
        HANDLE colour = GetStdHandle(STD_OUTPUT_HANDLE);
        SetConsoleTextAttribute(colour, 7);
        cout << "-----Passenger Info-----\n";
        InputName();
        InputGender();
    }
    virtual ~Passenger()
    {
        cout << "\n*****Passenger destructor called*****\n";
    }
};

class From_To :public Passenger
{
protected:

```

```

char destin;

char depart;

public:

void travel()
{
    HANDLE colour = GetStdHandle(STD_OUTPUT_HANDLE);
    SetConsoleTextAttribute(colour, 1);
    cout << "Enter \n 'I' for Islamabad \n";
    cout << "'K' for Karachi \n";
    cout << "'M' for Multan \n";
    cout << "'L' for Lahore";
    cout << "\n\t Departure from : ";
    cin >> depart;

    if (depart == 'i' || depart == 'I' || depart == 'l' || depart == 'L' || depart == 'k' || depart
== 'K' || depart == 'm' || depart == 'M');
    else
    {
        cout << "\n\n\t\t\tInvalid Selection";
        cout << "\n\t\t\t\t\tSelect Again :";
        cin >> depart;
    }

    HANDLE c = GetStdHandle(STD_OUTPUT_HANDLE);
    SetConsoleTextAttribute(c, 9);
    cout << "Enter \n";
    cout << "'I' for Islamabad \n";
    cout << "'K' for Karachi \n";
    cout << "'M' for Multan \n";
    cout << "'L' for Lahore";
    cout << "\n\t Destination : ";

```

```

        cin >> destin;

        if ((destin == 'i' || destin == 'I' || destin == 'l' || destin == 'L' || destin == 'k' || destin ==
'K' || destin == 'm' || destin == 'M') && (depart != destin));

        else

        {

            if (destin == depart)

                cout << "\n\nPlane Cannot Travel From " << depart << " back to " <<
destin;

                cout << "\n\n\t\tInvalid Selection";

                cout << "\n\t\t\t\tSelect Again : ";

                cin >> destin;

            }

        }

virtual ~From_To()

{

    cout << "\n*****From_To destructor called*****\n";

}

};

class AddBill :public From_To

{

protected:

    int bill = 0;

public:

    void Bill()

    {

        if ((depart == 'I' && destin == 'K') || (depart == 'K' && destin == 'I'))

            bill += 5000;

        if ((depart == 'I' && destin == 'M') || (depart == 'M' && destin == 'I'))

            bill += 4500;

    }

}

```

```

        if ((depart == 'I' && destin == 'L') || (depart == 'L' && destin == 'I'))
            bill += 3500;

        if ((depart == 'L' && destin == 'K') || (depart == 'K' && destin == 'L'))
            bill += 4500;

        if ((depart == 'L' && destin == 'M') || (depart == 'M' && destin == 'L'))
            bill += 3500;

        if ((depart == 'M' && destin == 'K') || (depart == 'K' && destin == 'M'))
            bill += 3000;

    }

    virtual ~AddBill()
    {
        cout << "\n*****AddBill destructor called*****\n";
    }
};

class Class : public AddBill
{
protected:
    int ClassType;

public:
    void InputClass()
    {
        HANDLE colour = GetStdHandle(STD_OUTPUT_HANDLE);
        SetConsoleTextAttribute(colour, 6);

        cout << "Press 1 for Second class\tcharges = 0 Rs\n";
        cout << "Press 2 for Economy class\tcharges = 2000 Rs\n";
        cout << "Press 3 for Busniss class\tcharges = 3000 Rs\n";
        cin >> ClassType;

        if ((ClassType != 1) && (ClassType != 2) && (ClassType != 3))
        {

```



```

        cout << "\a***invalid input enter again : ";
        cin >> ClassType;
    }
    Class::ADDtoBill();
}

void ADDtoBill()
{
    switch (ClassType)
    {
        case 1:
            bill += 0;
            break;
        case 2:
            bill += 2000;
            break;
        case 3:
            bill += 3000;
            break;
        default:
            cout << "Error";
            break;
    }
}

virtual ~Class()
{
    cout << "\n*****Class destructor called*****\n";
}

};

class TIME :public Class

```

```

{
protected:
    string date;
    char choice;
    int hour;
    int minute;
public:

    void Time()
    {
        HANDLE colour = GetStdHandle(STD_OUTPUT_HANDLE);
        SetConsoleTextAttribute(colour, 10);

    up1:
        cout << "Press Y if you want urgent ticket or Press N for advance booking...(Y/N).....\n";
        cout << "Enter in capital letter...";
        cin >> choice;
        if (choice == 'Y')
        {
            srand((unsigned)time(0));
            hour = (rand() % 12) + 1;
            srand((unsigned)time(0));
            minute = (rand() % 60) + 1;
            cout << "Train will leave at :: " << hour << " : " << minute << " " << "PM" << endl;
            bill += 500;
        }
        else if (choice == 'N')
        {
            cout << "Enter date you want to travel :: ";

```

```

        cin >> date;
    }
    else
    {
        cout << "\a*****invalid input";
        goto up1;
    }
}
virtual ~TIME()
{
    cout << "\n*****TIME destructor called*****\n";
}
};

class Num_of_Seats : public TIME
{
protected:
    int num;
public:
    Num_of_Seats()
    {
        num = 0;
    }
    void seats()
    {
up1:
        cout << "How much seats you want to book...? : ";
        cin >> num;
        if (num > 10)
        {

```

```

        cout << "\n\a*****you can get maximum 10 seats at a time \n";
        goto up1;
    }

    bill = bill * num;

    cout << "Ticket price : " << bill << endl;

}

virtual ~Num_of_Seats()
{
    cout << "\n*****Num_of_Seats destructor called*****\n";
}

};

class Ticket : public Num_of_Seats
{
public:
    void DisplayTicket()
    {
        HANDLE colour = GetStdHandle(STD_OUTPUT_HANDLE);
        SetConsoleTextAttribute(colour, 15);

        cout << "Passenger info\n";
        cout << "Passenger Name : " << NameOfPassenger;
        cout << "\nGender : " << PassengerGender;
        cout << "\nTravel from " << depart << " to " << destin << endl;
        cout << "Number of seats : " << num << endl;
        cout << "Ticket price + class charges : " << Class::bill << " Rs" << endl;
        if (choice == 'Y')
        {
            cout << "Train will leave at : " << hour << " : " << minute << " pm" << endl;
            cout << "+Extra charges : " << bill << " Rs" << endl;
        }
    }
}

```

```

    }

    else if (choice == 'N')
    {
        cout << "Train will leave on : " << date;
    }
}

virtual ~Ticket()
{
    cout << "\n*****Ticket destructor called*****\n";
}

};

class Writeonfile : public Ticket
{
public:
    void writeinfile()
    {
        ofstream ticket;
        ticket.open("Ticket.txt");
        ticket << "Passenger info\n";
        ticket << "Passenger Name : " << NameOfPassenger;
        ticket << "\nGender : " << PassengerGender;
        ticket << "\nTravel from " << depart << " to " << destin << endl;
        ticket << "Number of seats : " << num << endl;
        ticket << "Ticket price + class charges : " << Class::bill << " Rs\n";
        if (choice == 'Y')
        {
            ticket << "Train will leave at : " << hour << " : " << minute << " pm" << endl;
            ticket << "+Extra charges : " << bill << " Rs" << endl;
        }
    }
}

```

```

    }

    else if (choice == 'N')
    {
        ticket << "Train will leave on : " << date;
    }
}

virtual ~Writeonfile()
{
    cout << "\n*****Writeonfile destructor called*****\n";
}

};

class Menu : public Writeonfile
{
public:
    void DisplayMenu()
    {
        HANDLE colour = GetStdHandle(STD_OUTPUT_HANDLE);
        SetConsoleTextAttribute(colour, 13);
        cout << "\n\n\n\t*Main Menu*\n\n\n";
        cout << "\tPress 1 to enter ticket info\n\n";
        cout << "\tPress 2 to see ticket\n\n";
        cout << "\tPress 3 to see discription\n\n";
        cout << "\tPress 4 to exit\n\n";
        cout << "Enter choice : ";
    }

    ~Menu()
    {
        cout << "\n*****Menu destructor called*****\n";
    }
}

```

```

};

int main()
{
    Menu Object;

    bool End = false;

    while (!End) {
        system("cls");

        Object.WelcomeMessage();

        _getch();

    StartOfProgram:

        system("cls");

        char opt;

        Object.DisplayMenu();

        opt = _getch();

        switch (opt)
        {
            case '1':

                system("cls");

                cout << "Press Any Key to Start Booking\n";

                _getch();

                Object.InputPassenger();

                Object.travel();

                _getch();

                Object.Bill();

                system("cls");

                Object.InputClass();

                _getch();

                system("cls");

                Object.Time();

```

```

        Object.seats();

        cout << "\nTicket Generated...\nPress Any key to return to menu\n";

        _getch();

        goto StartOfProgram;

        break;

    case '2':

        system("cls");

        Object.DisplayTicket();

        Object.writeinfile();

        cout << "\n\nPress ANY Key to Return to Menu\n";

        _getch();

        goto StartOfProgram;

        break;

    case '3':

        system("cls");

        Object.display();

        _getch();

        goto StartOfProgram;

        break;

    case '4':

        _Exit(4);

        break;

    default:

        cout << "error";

        break;

    }

}

return 0;

}

```


-----*End*-----