**OBJECT ORIENTED CONCEPT & PROGRAMMING**

**(SE-201) LAB-5**

**TAQI HAIDER\_CSIT\_SECTION:B\_ROLL#CT-22092**

**Exercise:-**

**Q1:-**

#include <iostream>

#include <string>

using namespace std;

class Flight

{

private:

    int flight\_no;

    string source;

    string destination;

    int available\_seats;

public:

    Flight() : flight\_no(0), available\_seats(0), source("empty"), destination("empty") {}

    Flight(int x, int y) : flight\_no(x), available\_seats(y) {}

    void set\_data()

    {

        cout << "Enter Source: ";

        cin >> source;

        cout << "Enter Destination: ";

        cin >> destination;

        lengthCheck();

    }

    void reserve\_seat(int seat)

    {

        if (seat > available\_seats)

        {

            cout << "Seats not available!" << endl;

        }

        else

        {

            available\_seats -= seat;

            cout << "Your seats have been reserved!" << endl;

        }

    }

    void cancel\_reservation(int seat, char choice)

    {

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

        {

            int temp;

            cout << "Number of seats you want to cancel: ";

            cin >> temp;

            if (temp > seat)

            {

                cout << "Invalid Input!" << endl;

            }

            else

            {

                available\_seats += temp;

                cout << "You have canceled " << seat - temp << " seats" << endl;

                cout << "Available Seats: " << available\_seats << endl;

            }

        }

        else

        {

            cout << "Present Seats Available are " << available\_seats << endl;

        }

    }

    void display()

    {

        cout << "Flight Number: " << getFlightNumber() << endl;

        cout << "From: " << getSource() << endl;

        cout << "To: " << getDestination() << endl;

    }

    int getFlightNumber() const

    {

        return flight\_no;

    }

    string getSource() const

    {

        return source;

    }

    string getDestination() const

    {

        return destination;

    }

    pair<int, int> getFlightInfo() const

    {

        return make\_pair(flight\_no, available\_seats);

    }

    void lengthCheck()

    {

        if (source.length() > 3)

        {

            source = source.substr(0, 3);

            for (char &c : source)

            {

                c = toupper(static\_cast<unsigned char>(c));

            }

        }

        if (destination.length() > 3)

        {

            destination = destination.substr(0, 3);

            for (char &c : destination)

            {

                c = toupper(static\_cast<unsigned char>(c));

            }

        }

    }

};

int main()

{

    Flight f[2];

    int flight\_no[2], available\_seats;

    for (int i = 0; i < 2; i++)

    {

        cout << "Enter Details of Flight " << i + 1 << endl;

        cout << "Enter Flight Number: ";

        cin >> flight\_no[i];

        cout << "Enter Number Of Seats Available: ";

        cin >> available\_seats;

        f[i] = Flight(flight\_no[i], available\_seats);

        f[i].set\_data();

        int seat;

        cout << "Enter Number Of Seats You Want To Reserve: ";

        cin >> seat;

        f[i].reserve\_seat(seat);

        char choice;

        cout << "Do you want to modify/cancel your reservations (Y/y or N/n): ";

        cin >> choice;

        f[i].cancel\_reservation(seat, choice);

        f[i].display();

    }

    cout << endl;

    for (int i = 0; i < 2; i++)

    {

        if (flight\_no[i] == flight\_no[i + 1])

        {

            cout << "The Flights Are Same!" << endl;

        }

        else

        {

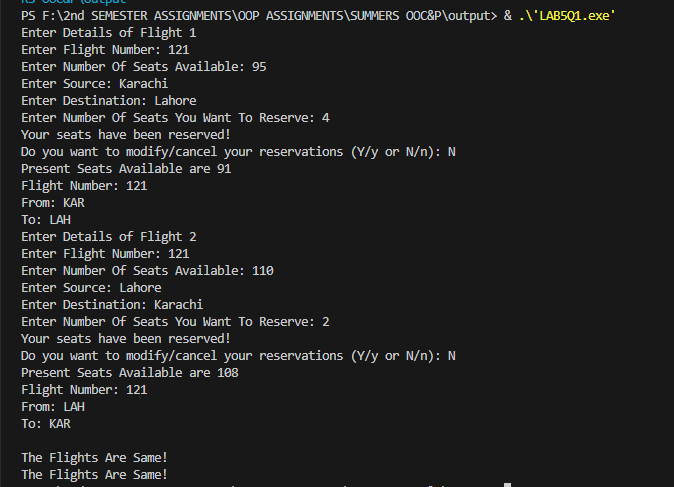
            cout << "The Flights Are Not Same!" << endl;

        }

    }

    return 0;

}



**Q2:-**

#include <iostream>

using namespace std;

class Car{

private:

    string brandName;

    float price\_new;

    string color;

    float odometer;

    float temp;

public:

    car(){

        cout << "Enter Brand Name: ";

        cin >> brandName;

        cout << "Enter Price When The Car Was New: ";

        cin >> price\_new;

        cout << "Enter Color: ";

        cin >> color;

        cout << "Enter Current Odometer Reading: ";

        cin >> odometer;

    }

    float getPrice\_afterUse(){

        temp = price\_new \* (1 - (odometer / 600000));

        return temp;

    }

    void updateMilage(double traveled\_distance){

        double milage = traveled\_distance + odometer;

        odometer = milage;

    }

    void display(){

        cout << "Brand Name: " << brandName << endl;

        cout << "New Price: " << price\_new << endl;

        cout << "Price After Used: " << temp << endl;

        cout << "Odometer: " << odometer << endl;

    }

};

int main(){

    Car A;

    A.getPrice\_afterUse();

    cout << endl;

    A.display();

    cout << endl;

    for (int i = 0; i < 2; i++){

        double distance[2];

        cout << "Enter Distance Travelled: ";

        cin >> distance[i];

        A.updateMilage(distance[i]);

        A.getPrice\_afterUse();

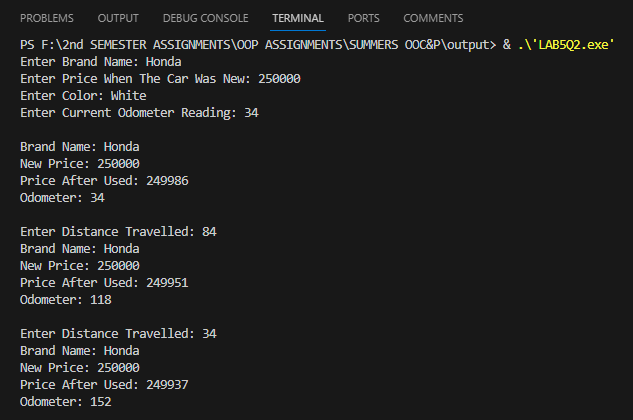
        A.display();

        cout << endl;

    }

    return 0;

}



**Q3 :-**

#include <iostream>

using namespace std;

class coffee\_outlet

{

private:

    int no\_of\_bags;

    float cost\_of\_order;

    int largeBoxes;

    int mediumBoxes;

    int smallBoxes;

    float const bag\_price = 5.50;

    float largeB;

    float mediumB;

    float smallB;

    float total\_cost;

public:

    coffee\_outlet()

    {

        cout << "Enter Number Of Bags Orderd:  ";

        cin >> no\_of\_bags;

    }

    void costOf\_order()

    {

        cost\_of\_order = bag\_price \* no\_of\_bags;

    }

    void no\_of\_boxes()

    {

        largeBoxes = no\_of\_bags / 20;

        mediumBoxes = (no\_of\_bags % 20) / 10;

        smallBoxes = ((no\_of\_bags % 20) % 10) / 5;

    }

    void priceForboxes()

    {

        largeB = largeBoxes \* 1.80;

        mediumB = mediumBoxes \* 1.00;

        smallB = smallBoxes \* 0.60;

        total\_cost = cost\_of\_order + largeB + mediumB + smallB;

    }

    void display()

    {

        cout << "The Cost Of Order: $ " << cost\_of\_order << endl;

        cout << endl;

        cout << "BOX Used: " << endl;

        cout << "\t" << largeBoxes << " Large"

             << "- $ " << largeB << endl;

        cout << "\t" << mediumBoxes << " Medium"

             << "- $ " << mediumB << endl;

        cout << "\t" << smallBoxes << " Small"

             << "- $ " << smallB << endl;

        cout << endl;

        cout << "Your total cost is: $  " << total\_cost << endl;

    }

};

int main()

{

    coffee\_outlet c1;

    c1.costOf\_order();

    c1.no\_of\_boxes();

    c1.priceForboxes();

    c1.display();

    return 0;

}

