Lab 8 Tasks

Hamza Farhan

24K-0576

Task 1

#include <iostream>

#include <string>

using namespace std;

class Humidity;

class Temperature{

    private:

    float temp;

    public:

    Temperature(float t):temp(t){}

    friend float calculateHeatIndex(Temperature obj,Humidity ob);

};

class Humidity{

    private:

    float hum;

    public:

    Humidity(float h):hum(h){}

    friend float calculateHeatIndex(Temperature obj,Humidity ob);

};

float calculateHeatIndex(Temperature obj,Humidity ob){

    return obj.temp +(0.05\*ob.hum);

}

int main() {

    Temperature temp(32.0);

    Humidity hum(70.0);

    float heatIndex = calculateHeatIndex(temp, hum);

    cout << "Calculated Heat Index: " << heatIndex << endl;

    return 0;

}

Output



Task 2

#include <iostream>

#include <string>

using namespace std;

class Librarian;

class Book{

    private:

    string title;

    float price;

    public:

    Book(string t,float p):title(t),price(p){}

    friend Librarian;

};

class Librarian{

    public:

    void applyDiscount(Book b){

        cout<<"applying discount\nprice before discount: "<<b.price<<endl;

        b.price\*=0.7;

        cout<<"after 30% discount: "<<b.price<<"\n";

    }

    void displayInfo(Book b){

        cout<<"title: "<<b.title<<endl;

        cout<<"price: "<<b.price<<endl;

    }

};

int main (){

    Book b("harry potter",1000);

    Librarian l;

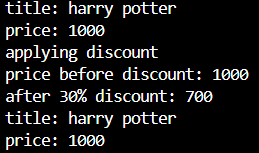
    l.displayInfo(b);

    l.applyDiscount(b);

    l.displayInfo(b);

}

Output



Task 3

#include <iostream>

#include <string>

using namespace std;

class Teacher;

class Student{

    private:

    string name;

    int grades[3];

    public:

    Student(string n,int g[]):name(n){

        for (int i = 0; i <3; i++)

        {

            grades[i]=g[i];

        }

    }

    friend  Teacher;

    friend float calculateAverageGrade(Student s);

};

class Teacher{

public:

    void update(Student s){

        int g,i;

        cout<<"enter new grade: ";

        cin>>g;

        cout<<"enter subject index you want to change: ";

        cin>>i;

        s.grades[i]=g;

    }

    void viewgrade(Student s){

        cout<<s.grades[0]<<endl;

        cout<<s.grades[1]<<endl;

        cout<<s.grades[2]<<endl;

    }

};

float calculateAverageGrade(Student s){

    float avg=0;

    for (int i = 0; i < 3; i++)

    {

        avg+=s.grades[i];

    }

    avg=avg/3;

    return avg;

}

int main(){

    int g[3]={90,95,100};

    Student s("hamza",g);

    Teacher t;

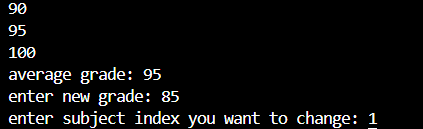
    t.viewgrade(s);

    cout<<"average grade: "<<calculateAverageGrade(s)<<endl;

    t.update(s);

}

Output



Task 4

#include <iostream>

#include <string>

using namespace std;

class Currency{

    public:

    float f;

    Currency(float f):f(f){}

    Currency operator+(Currency c){

        return Currency(f+c.f);

    }

    Currency operator-(){

        return Currency(-f);

    }

    Currency operator-(Currency c){

        return Currency(f-c.f);

    }

    Currency& operator+=(const Currency& c) {

        f += c.f;

        return \*this;

    }

    Currency& operator-=(const Currency& c) {

        f -= c.f;

        return \*this;

    }

    friend ostream& operator<<(ostream& os,Currency c);

};

ostream& operator<<(ostream& os,Currency c){

    os<<c.f;

    return os;

}

int main() {

    Currency p1(19.99);

    Currency p2(5.49);

    Currency d(2.50);

    cout << "initial values:\n";

    cout << "P1: " << p1 << "\nP2: " << p2 << "\nD: " << d << "\n\n";

    Currency sum = p1 + p2;

    cout << "P1 + P2 = " << sum << endl;

    Currency diff = p1 - p2;

    cout << "P1 - P2 = " << diff << endl << endl;

    cout << "negative of P1: " << -p1 << endl;

    cout << "applying discount (-" << d << ")" << endl;

    p1 -= d;

    cout << "new P1: " << p1 << endl;

    p2 += d;

    cout << "P2 after adding discount: " << p2 << endl;

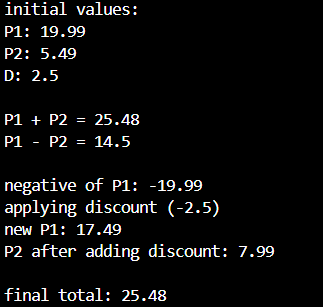
    sum = p1 + p2;

    cout << "\nfinal total: " << sum << endl;

    return 0;

}

Output



Task 5

#include <iostream>

#include <string>

using namespace std;

class Fraction {

private:

    int num;

    int den;

public:

    Fraction(int n = 0, int d = 1) {

        num = n;

        den = d;

    }

    Fraction operator+( Fraction f){

        int n=num\*f.den+f.num\*den;

        int d=den\*f.den;

        return Fraction(n, d);

    }

    Fraction operator-( Fraction f){

        int n=num\*f.den-f.num \*den;

        int d=den\*f.den;

        return Fraction(n, d);

    }

    Fraction operator\*(Fraction f){

        return Fraction(num\*f.num,den\*f.den);

    }

    Fraction operator/( Fraction f){

        return Fraction(num\*f.den,den\*f.num);

    }

    // Overload <<

    friend ostream& operator<<(ostream& os,  Fraction f){

        os << f.num << "/" << f.den;

        return os;

    }

};

int main() {

    Fraction f1(3, 4);

    Fraction f2(5, 6);

    cout << "f1 = " << f1 << endl;

    cout << "f2 = " << f2 << endl << endl;

    cout << "f1 + f2 = " << (f1 + f2) << endl;

    cout << "f1 - f2 = " << (f1 - f2) << endl;

    cout << "f1 \* f2 = " << (f1 \* f2) << endl;

    cout << "f1 / f2 = " << (f1 / f2) << endl;

    return 0;

}

Output

