

ÖDEV 2:

CEVAPLAR

1. [Operatör Yükleme, Operator Overloading] Bu kısımda, 1nci ödevde 5nci sorudaki programa (Aşağıda bu program verilmektedir) ilave fonksiyonlar yazmanız beklenmektedir. Ayrıca main fonksiyonu içerisinde bu operatör fonksiyonlarının doğru çalıştığını gösteren test kodları yazmalısınız.

```
#include <iostream>
#include <string>
using namespace std;

struct Product{
    string name;
    double price;
    int quantity;
    double tax; //this rate should be between 0 and 1
};

namespace Vendor1{

    double totalPrice(const Product &p){

        double price=p.price*p.quantity;
        return price+price*p.tax;
    }

    void compare(const Product &p1, const Product &p2){

        double unitPrice1=p1.price+p1.price*p1.tax;
        double unitPrice2=p2.price+p2.price*p2.tax;

        if(p1.name==p2.name){

            if(unitPrice1>unitPrice2)
                cout<<p1.name
                 <<" is more expensive than "<<p2.name<<endl;
            else if (unitPrice1==unitPrice2)
                cout<<p1.name
                 <<" has the same price as "<<p2.name<<endl;
            else
                cout<<p1.name<<" is cheaper than "<<p2.name<<endl;
        }
        else{
            cout<<"These are not the same product. "<<endl;
        }
    }

};
```

```

namespace Vendor2{
    double totalPrice(const Product &p){

        return p.price*p.quantity;
    }

    void compare(const Product &p1, const Product &p2){

        if(p1.name==p2.name){

            if(p1.price>p2.price)
                cout<<p1.name
                 <<" is more expensive than "<<p2.name<<endl;
            else if (p1.price==p2.price)
                cout<<p1.name
                 <<" has the same price as "<<p2.name<<endl;
            else
                cout<<p1.name<<" is cheaper than "<<p2.name<<endl;

        }
        else{
            cout<<"These are not the same product. "<<endl;
        }
    }
};

int main(){

    Product testProduct1={"book",25.00,4,.50};
    Product testProduct2={"book",30.00,2,.10};
    Product testProduct3={"pencil",5.5,2,.10};

    // TODO: Write the statement such that testProduct1 and testProduct2
    //         will be compare by using the function provided by Vendor 1
    Vendor1::compare(testProduct1,testProduct2);

    // TODO: Write the statement such that testProduct1 and testProduct2
    //         will be compare by using the function provided by Vendor 2
    Vendor2::compare(testProduct1,testProduct2);

    // TODO: Write the statement such that testProduct1 and testProduct3
    //         will be compare by using the function provided by Vendor 1
    Vendor1::compare(testProduct1,testProduct3);

    // TODO: Write the statement such that the total price of testProduct1
    //         will be calculated by using the function provided by Vendor 2
    cout<<"The total price of testProduct1 is "
         <<Vendor2::totalPrice(testProduct1)<<endl;

    // TODO: Write the statement such that the total price of testProduct1
    //         will be calculated by using the function provided by Vendor 1
    cout<<"The total price of testProduct1 is "
         <<Vendor1::totalPrice(testProduct1)<<endl;
}

```

- (a) == operator fonksiyonu ekleyiniz. Bu operator kullanıldığında, iki product veri yapısı için name, price, quantity, tax değerlerinin hepsinin eşit olmasına bakacaktır. Hepsi eşitse, true değilse false döndürecek.

Örneğin;

```
Product p1,p2;
if (p1==p2)
    ...
```

Cevap:

```
/* Equality operator of Product type objects */
bool operator==(const Product& op1, const Product& op2) {
    // Check names are equal or not
    if (op1.name != op2.name)
        return false;
    // Check prices are equal or not
    if (op1.price != op2.price)
        return false;
    // Check quantities are equal or not
    if (op1.quantity != op2.quantity)
        return false;
    // Check taxes are equal or not
    if (op1.tax != op2.tax)
        return false;
    // return if all of them are equal
    return true;
}
```

- (b) ++ operator fonksiyonun prefix ve postfix versiyonlarını ekleyiniz. Bu operator kullanıldığında, product veri yapısının quantity değeri 1 arttırılacaktır.

Örneğin;

```
Product p1;
p1++; //posfix
++p1; //prefix
    ...
```

Cevap:

```
/* Prefix increment operator of Product type objects */
Product operator++(Product& op) {
    // Increase quantity
    op.quantity++;
    return op;
}

/* Postfix increment operator of Product type objects */
Product operator++(Product& op, int val) {
    Product tmp=op;
    // Increase quantity
    op.quantity++;
    return tmp;
}
```

- (c) + operator fonksiyonu yazınız. Operatorün bir tarafında product veri yapısı ve bir tarafında tam sayı varken, product veri yapısının quantity değeri tam sayı kadar arttırılacaktır.

Örneğin;

```
Product p1,p2;
p1=p2+5;
p1=5+p2;
...
```

Cevap:

```
/* Right side integer addition operator of Product type
objects */
Product operator+(const Product& op, int amount) {
    // Copy the argument object to a locally created object
    Product p = op;
    // Increase quantity by given amount
    p.quantity += amount;
    // Return a copy of locally created object
    return p;
}
/* Left side integer addition operator of Product type
objects */
Product operator+(int amount, const Product& op) {
    // Copy the argument object to a locally created object
    Product p = op;
    // Increase quantity by given amount
    p.quantity += amount;
    // Return a copy of locally created object
    return p;
}
```

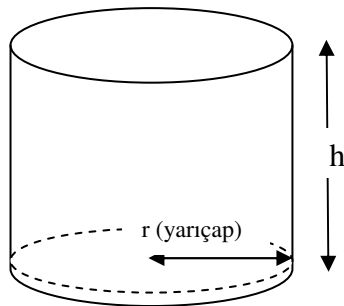
2. [Sınıflar ve nesneler, Classes and objects] Bu kısımda, aşağıda istenenlere göre program yazınız.

- Aşağıda UML diyagramı verilen Silindir sınıfını kodlayınız.
- Ayrıca, Silindir nesnesinin kullanıldığı bir uygulama yazınız. Uygulama çalıştırıldığında aşağıda gösterilen örnek çıktı alınabilecektir.

Silindir
- yükseklik: int - yarıcap: int
+ setParameters(h: int, r: int) + yuzeyAlaniHesapla(): double + hacimHesapla(): double

Yüzey alanı: $A = 2\pi rh + 2\pi r^2$

Hacim: $H = \pi r^2 h$



Yazılacak programdan aşağıdaki gibi çıktı alınabilecektir.

Örnek çıktı: (Koyu karakterle gösterilenler kullanıcı tarafından girilecek değerlerdir.)

```
Yukseklık (h): 5↵
Taban yarıçapı (r): 3↵
-----
Yuzeı alanı (A): 150.756
Hacmi (H): 141.372
```

Cevap:

(a)

```
// Define a class called Silindir
```

```
#define PI 3.14166
```

```
class Silindir {
    /* Public Scope of the class*/
public:
    // Parameter setter
    void setParameters(int h, int r){
        // Set height of the cylinder to given argument
        yukseklik = h;
        // Set radius of the cylinder to given argument
        yarıcap = r;
    } // end of the setParameters implementation

    // Surface area calculator
    double yuzeyAlaniHesapla(){
        // Define a local variable for the area
        double area;
        // Calculate the area
        area=2*PI*yarıcap*yukseklik + 2 * PI* yarıcap *yarıcap;
        // return area
    }
}
```

```

        return area;
    }

    // Volume calculator
    double hacimHesapla(){
        // Define local variable for the volume
        double volume;
        // Calculate the volume
        volume = PI*yaricap*yaricap*yukseklik;
        // return volume
        return volume;
    }

    /* Private Scope*/
private: // Private variable for the height of a cylinder
        int yukseklik; // Private variable for the radius of a
                        // cylinder
        int yaricap;
}; // end of the class definition

```

(b)

```

// SilindirApp.cpp
#include <iostream>
using namespace std;

int main() {
    int h, r;
    cout << "Yukseklik (h) : ";
    cin >> h;
    cout << "Taban yaricapi (r) : ";
    cin >> r;
    cout << "-----" << endl;
    Silindir s;
    s.setParameters(h, r);
    cout<<"Yuzey alani (A) : "<<s.yuzeyAlaniHesapla()<< endl;
    cout << "Hacmi (H) : " << s.hacimHesapla() << endl;
}

```