

ÖDEV 1:**CEVAPLAR**

1. [Kapsam, Scope] Aşağıdaki soruları cevaplayınız.

(a) Aşağıda verilen programa göre tabloyu doldurunuz. Tablodaki boşluklara E ya da H harfleri yazılacaktır. Belirtilen değişken belirtilen alan içinde görünür ise Evet (E) , değilse Hayır (H) harfi yazılacaktır.

<pre>#include <iostream> using namespace std; const double RATE = 10.50; int z; double t; void one(int x, char y); void two(int a, int b, char x); void three(int one, double y, int z); int main(){ int num, first; double x, y, z; char name, last; . . . return 0; } void one(int x, char y) { . . . } int w;</pre> <p><i>Devami sağdaki sütunda...</i></p>	<p><i>Programın devamı...</i></p> <pre>void two(int a, int b, char x) { int count; . . . } void three(int one, double y, int z) { char ch; int a; . . . //Block four { int x; char a; . . } //end Block four . . . }</pre>
--	--

Cevap:

Değişken	One fonksiyonu içinde görünürlüğü	Two fonksiyonu içinde görünürlüğü	Three fonksiyonu içinde görünürlüğü	Block four içinde görünürlüğü	Main fonksiyonu içinde görünürlüğü
RATE	√	√	√	√	√
z (main'den önce)	√	√	√	√	√
t (main'den önce)	√	√	√	√	√
Main lokal değişkenleri					√
one (fonksiyon adı)	√	√	√	√	√

x (one fonksiyonunun formal parametresi)	√				
w (two fonksiyonundan önce)		√	√	√	
two (fonksiyon adı)	√	√	√	√	√
a (two fonksiyonunun formal parametresi)		√			
Two lokal değişkenleri		√			
three (fonksiyon adı)	√	√	√	√	√
one (three fonksiyonunun formal parametresi)			√	√	
ch (three lokal değişkeni)			√	√	
x (Block four lokal değişkeni)				√	
a (Block four lokal değişkeni)				√	

(b) Aşağıdaki programın çıktısını yazınız.

```
#include <iostream>
int x = 2;

int main()
{
    int x = 7;
    std::cout << "\nStarting Program\n";
    {
        int x = 5; // declare new variable
        std::cout << "x = " << x << '\n';
        {
            int x = 9;
            std::cout << "x = " << x << '\n';
        }
        std::cout << "x = " << x << '\n';
        {
            x = 1;
        }
        std::cout << "x = " << x << '\n';
    }
}
```

Cevap:

```
Starting Program
x = 5
x = 9
x = 5
x = 1
```

2. [İsin Alanları, Namespaces] Aşağıdaki programı inceleyiniz.

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

namespace V1 {
    void f() {cout<<"V1::f"<<endl;}
    void g() {cout<<"V1::g"<<endl;}
}

namespace V2 {
    void f() {cout<<"V2::f"<<endl;}
    void g() {cout<<"V2::g"<<endl;}
    void k() {cout<<"V2::k"<<endl;}
}

void h() {
    using namespace V2;
    using V1::f;
    f();
    g(); //A
    V2::f();
    k();
}

using namespace V1;

int main(){
    h();
    f(); //B
    V2::g();
    return 0;
}
```

Programın çıktısı nedir? Derleyici kullanarak, cevabınızı kontrol ediniz. Program çıktısı üzerinde, //A ve //B bulunan satırlarda ekrana yazdırılan çıktıyı belirtiniz.

Cevap:

```
V1::f
V2::g    -> A
V2::f
V2::k
V1::f    -> B
V2::g
```

3. [Varsayılan Parametreler, Default Parameters]

(a) Aşağıdaki fonksiyon prototipine göre soruyu cevaplayınız.

```
void testDefaultParam(int a, int b = 7, char z = '*');
```

Aşağıdaki fonksiyon çağrılarından hangisi(leri) doğrudur? Açıklayınız.

- a. testDefaultParam(5);
- b. testDefaultParam(5, 8);
- c. testDefaultParam(6, '#');
- d. testDefaultParam(0, 0, '*');

Cevap:

Hepsi doğrudur.

(b) Aşağıdaki fonksiyona göre cevaplayınız.

```
void defaultParam(int u, int v = 5, double z = 3.2)
{
    int a;
    u = u + static_cast<int>(2 * v + z);
    a = u + v * z;
    cout << "a = " << a << endl;
}
```

Aşağıdaki fonksiyon çağrılarının çıktısı nedir? (Not: `static_cast` derste bahsedilmedi. İnternette araştırıp öğreniniz.)

- a. `defaultParam(6);`
- b. `defaultParam(3, 4);`
- c. `defaultParam(3, 0, 2.8);`

Cevap:

- a. **a = 35**
- b. **a = 26**
- c. **a = 5**

4. [Değer parametresi, Function-pass by value ve referans parametresi ,pass by reference]

(a) Aşağıdaki programın çıktısını yazınız. (Programı çalıştırdıktan sonra, 25/05/2006 şeklinde giriş yaptığınızı kabul ediniz.)

```
#include<iostream>
using std::cout;
using std::cin;
using std::endl;

//prototype functions
void readDate(int& month, int& day, int& year); //reference parameters
void printDate(int month, int day, int year); //value parameters

//initialize constants
//initialize global variables

int main()
{
    //initialize automatic variables
    int mm, dd, yy;
    readDate(mm, dd, yy);
    printDate(mm, dd, yy);

    return 0;
}

//function definitions
void readDate(int& month, int& day, int& year)
{
```

```
char ch; //local variable
cout << "Enter a date (mm/dd/year): ";
cin >> month >> ch >> day >> ch >> year;
}

void printDate(int month, int day, int year)
{
    cout << "The date is " << month << '-' << day;
    cout << '-' << year << endl;
}
```

Cevap:

Enter a date (mm/dd/year): 25/05/2006 ↵
The date is 25-5-2006

(b) Aşağıdaki programın çıktısını yazınız.

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

void find(int a, int& b, int& c,);

int main(){
    int one, two, three;
    one = 5;
    two = 10;
    three = 15;
    find(one, two, three);
    cout << one << ", " << two << ", " << three << endl;
    find(two, one, three);
    cout << one << ", " << two << ", " << three << endl;
    find(three, two, one);
    cout << one << ", " << two << ", " << three << endl;
    find(two, three, one);
    cout << one << ", " << two << ", " << three << endl;
    return 0;
}

void find(int a, int& b, int& c)
{
    int temp;
    c = a + b;
    temp = a;
    a = b;
    b = 2 * temp;
}
```

Cevap:

5, 10, 15
20, 10, 15
25, 30, 15
45, 30, 60

5. Aşağıdaki programda, TODO kısmındaki açıklamalara uygun şekilde bu kısımlarda bulunması gereken kodları yazınız. Daha sonra programın çıktısını yazınız.

```
#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
};

// TODO: Write the statements such that the following two functions will be
//        covered by the namespace of 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;
    }
}

// TODO: Write the statements such that the following two functions will be
//        covered by the namespace of 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

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

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

    // TODO: Write the statement such that the total price of testProduct1
    //         will be calculted by using the function provided by Vendor 2

    // TODO: Write the statement such that the total price of testProduct1
    //         will be calculated by using the function provided by Vendor 1
}
```

Cevap:

```
#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
};

// TODO: Write the statements such that the following two functions will be
//         covered by the namespace of Vendor1
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;
        }
    }

};

// TODO: Write the statements such that the following two functions will be
//        covered by the namespace of Vendor2
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;
}
```