

## QUIZ 1: (Cevaplar)

## A. Aşağıdaki soruları verilen programa göre cevaplayınız:

```

namespace Point2D {
    struct Point {
        double x, y;
    };
    Point subtract(const Point & v1, const Point
&v2)
    {
        Point tmp;
        tmp.x = v1.x - v2.x;
        tmp.y = v1.y - v2.y;
        return tmp;
    }
    Point scalarProduct(Point & v1, double a)
    {
        v1.x = a*v1.x;
        v1.y = a*v1.y;
        return v1;
    }
    void setPoint(Point &v, double x=0, double y=0)
    {
        v.x = x;
        v.y = y;
    }
}
namespace Point3D {
    struct Point {
        double x, y, z;
    };
    Point scalarProduct(Point v1, double a)
    {
        v1.x = a*v1.x;
        v1.y = a*v1.y;
        v1.z = a*v1.z;
        return v1;
    }
    void setPoint(Point &v, double x, double y, double z
)
    {
        v.x = x;
        v.y = y;
        v.z = z;
    }
}
using namespace Point2D;
int main() {
    Point2D::Point p2d_1, p2d_2;
    Point3D::Point p3d_1, p3d_2;
    setPoint (p2d_1, 3);
    setPoint (p2d_2, 1, 2);
    Point3D::setPoint (p3d_1, 3, 4, 6);
    p2d_1 = subtract(p2d_1, p2d_2);
    p2d_2 = scalarProduct(p2d_1, 2);
    p3d_2 = Point3D::scalarProduct(p3d_1, 3);
    return 0;
}

```

(1) [30p] Namespace Point2D altındaki kodları sınıf olarak tekrar yazınız. main() fonksiyonunu da yaptığınız bu değişikliğe uygun olarak tekrar yazınız.

Cevap:

```

namespace Point2D {
    class Point {
    public:
        Point subtract(const Point &v) {
            Point tmp;
            tmp.x = x - v.x;
            tmp.y = y - v.y;
            return tmp;
        }
        Point scalarProduct(double a) {
            x = a*x;
            y = a*y;
            return (*this);
        }
        //this pointeri gösterilmediği için
        //bunu yapmayandan puan kırılmayacaktır.
        void setPoint(double _x = 0, double _y = 0) {
            x = _x;
            y = _y;
        }
    private:
        double x, y;
    };
}
namespace Point3D {
    class Point {
    public:
        Point scalarProduct(double a) {
            x = a*x;
            y = a*y;
            z = a*z;
            return (*this);
        }
        //this pointeri gösterilmediği için
        //bunu yapmayandan puan kırılmayacaktır.
        void setPoint(double _x, double _y, double _z) {
            x = _x;
            y = _y;
            z = _z;
        }
    private:
        double x, y, z;
    };
}
using namespace Point2D;
int main() {
    Point2D::Point p2d_1, p2d_2;
    Point3D::Point p3d_1, p3d_2;
    p2d_1.setPoint(3);
    p2d_2.setPoint(1, 2);
    p3d_1.setPoint(3, 4, 6);
    p2d_1 = p2d_1.subtract(p2d_2);
    p2d_2 = p2d_1.scalarProduct(2);
    p3d_2 = p3d_1.scalarProduct(3);
    return 0;
}

```

}

(2) [15p]

Yukarıda verilen program çalıştığında, main() fonksiyonunda “return 0;” kod satırında programı durdurulduğunda, aşağıdaki değerler nedir?

Cevap:

	x	y	z
p2d_1	4	-4	--
p2d_2	4	-4	--
p3d_1	3	4	6
p3d_2	9	12	18

(3) [15p]

namespace tanımlamaları kullanılmasaydı ne olurdu? Açıklayınız.

Cevap:

Birden çok aynı isimde yapı (struct) ve aynı prototipe sahip birden fazla fonksiyon bulunmaktadır. Uygulamada kullanılırken, hangi fonksiyon ya da yapının çağrılacağı belirsizlik içerir. Namespace tanımlamaları bu belirsizliği ortadan kaldırmaktadır.

(4) [15p]

Yukarıda verilen programda,  
p2d\_1 = subtract(p2d\_1, p2d\_2);  
yerine  
p2d\_1 = p2d\_1 - p2d\_2;  
yazabilmemiz için gereken fonksiyonu yazınız.

Cevap:

```
Point operator-(const Point& v1,const Point &v2)
{
    Point tmp;
    tmp.x = v1.x + v2.x;
    tmp.y = v1.y + v2.y;
    return tmp;
}
```

## B. [25p]Aşağıdaki programın çıktısını yazınız.

<pre>#include &lt;iostream&gt; using namespace std; class Product { public:     void setProduct(double, int);     void setUnitPrice(double p) {         unitPrice = p;     }     void setAmount(int);     double getPrice(); private:     double unitPrice;     int amount; }; void Product::setAmount(int a) {     if (a &gt;= 0)         amount = a;     else         amount = 0; } void Product::setProduct(double p, int a) {     setUnitPrice(p);     setAmount(a); }</pre>	<pre>double Product::getPrice() {     return (unitPrice*amount); }  int main() {     Product chocolate,tea;     chocolate.setProduct(3.5,2);     tea.setProduct(3, 7);     cout &lt;&lt; "Chocolate is "         &lt;&lt; chocolate.getPrice()         &lt;&lt; " TL" &lt;&lt; endl;     cout &lt;&lt; "Tea is " &lt;&lt; tea.getPrice()         &lt;&lt; " TL" &lt;&lt; endl;     cout &lt;&lt; "Total is "         &lt;&lt; chocolate.getPrice()+tea.getPrice()         &lt;&lt; " TL" &lt;&lt; endl; }  <b>Cevap:</b> Chocolate is 4.5 TL Tea is 10 TL Total is 14.5 TL</pre>
--	--