



CPP Module 02

Static veri elemanları (değişkenler)

Sınıf içinde normal değişken tanımladığımızda, bu sınıftan türeteceğimiz her bir nesne içinde bu değişken için farklı bir değer oluşturulmuş olur. Ancak, bir sınıf içinde tanımladığımız değişkeni static olarak tanımlarsak, bu sınıftan üreteceğimiz tüm nesneler bu değişken için verilen sadece tek bir değeri paylaşabilir ve kullanabilirler. Sınıftan ilk nesne oluşturulmadan önce, tüm static değişkenlere ilk değer olarak 0 (sıfır) değeri verilir.

Sınıf içinde static değişken tanımladığımızda, tüm nesneler aynı değeri kullanacağından, sınıf dışında programın herhangi bir yerinde, sınıf adı ile birlikte :: işlemcisi kullanarak, global olarak static değişkenler için bir tanımlama yapmamız gerekir.

Copy Constructor

- Öyle durumlar var ki, bir nesne hayata değerini başka bir nesneden alarak başlıyor. Böylesi durumlarda hayata gelen nesne için çağrılan constructor a COPY CONSTRUCTOR deniliyor.

Peki Copy Constructor Yapısı Nasıl?

- Copy Constructor kullanımını örnekte görmüş olduk. Biraz açıklayalım. Sınıf ismi ile aynı fakat diğer nesneye de ulaşmamız gerekiyor. Nesneye ulaşmam için onu referans yoluyla almam gerekiyor ve nesneyi okuma amacıyla kullanacağım için const olması gerekiyor. Fonksiyonumuzun yapısıda böyle.



<https://www.geogebra.org/m/JMMKv7cx>

A copy constructor is a member function that initializes an object using another object of the same class. In simple terms, a constructor which creates an object by initializing it with an object of the same class, which has been created previously is known as a copy constructor.

The process of initializing members of an object through a copy constructor is known as copy initialization.

```
class A final
{
    public:
        A (); //constructor
        A (const A &a); //copy constructor
        ~A (); //destructor
        A & operator = (const A &a); //copy assignment operator
};
```



<https://embeddedartistry.com/blog/2018/07/12/simple-fixed-point-conversion-in-c/>

To convert from floating-point to fixed-point, we follow this algorithm:

1. Calculate `x = floating_input * 2^(fractional_bits)`
2. Round `x` to the nearest whole number (e.g. `round(x)`)
3. Store the rounded `x` in an integer container



<https://learn.microsoft.com/en-us/cpp/cpp/operator-overloading?view=msvc-170>

The operator keyword declares a function specifying what the operator symbol means when applied to instances of a class. This gives the operator more than one meaning, or "overloads" it. The compiler distinguishes between the different meanings of an operator by examining the types of its operands.

ex02



<https://learn.microsoft.com/en-us/cpp/cpp/increment-and-decrement-operator-overloading-cpp?view=msvc-170>

When you write overloaded operator functions, it can be useful to implement separate versions for the prefix and postfix versions of these operators. To distinguish between the two, the following rule is observed: The prefix form of the operator is declared exactly the same way as any other unary operator; the postfix form accepts an extra argument of type `int`.



<https://www.cuemath.com/geometry/area-of-triangle-in-coordinate-geometry/>

$$(\Delta ABC) = (1/2) |x_1(y_2 - y_3) + x_2(y_3 - y_1) + x_3(y_1 - y_2)|$$

<https://www.youtube.com/watch?v=HYAgJN3x4GA>

$$w_1 = \frac{A_x(C_y - A_y) + (P_y - A_y)(C_x - A_x) - P_x(C_y - A_y)}{(B_y - A_y)(C_x - A_x) - (B_x - A_x)(C_y - A_y)}$$

$$w_2 = \frac{P_y - A_y - w_1(B_y - A_y)}{C_y - A_y}$$

Point P is inside triangle ABC if:

$$w_1 \geq 0$$

and

$$w_2 \geq 0$$

and

$$(w_1 + w_2) \leq 1$$