

2023 08 11

Classes

```
class members  
    data members  
        static data members  
        non-static data members  
    member function  
        static member function  
        non-static member function  
    member type / type member / nested member  
    class names
```

```
class MyClass {  
    int x ; // data member  
    void func(); // member function  
    class MyClass1; // member type  
};
```

```
class MyClass {  
    int x, y, z;  
};  
  
int main()  
{  
    // sizeof(Myclass) = 12  
    std::cout << "sizeof(Myclass) = " << sizeof(Myclass) << "\n";  
}
```

```
member selection operators  
    . ve ->  
        MyClass m;  
        m.x;  
        MyClass *p = &m;  
        p->x;
```

```
access spefier  
    private members  
    public members  
    protected members  
  
class'ta herhangi bir access spefier kullanmazsak private,  
struct'ta ise public olarak default olur  
  
private, public, protected ayrı scope değildir!
```

```

class Fighter{

    public:          //Fighter*
        void func(          );
}
int main(){

    Fighter f;
    f.func(); // burda aslında func gizli bir arguman alıyor o da f adresi
}

```

```

//member fonksiyonları 2 defa tanımlayamıyoruz
class Nec {
    public:
        //syntax hatası
        int foo(int);
        int foo(int);
};

```

```

class Nec{

private:
    int foo(int);
public:
    int foo(double);
};

int main(){
    Nec myNec;
    myNec.foo(12);
    //Hata döner çünkü önce hangi fonksiyonun çağrılacağı seçimi yapılıyor sonra erişim yapılıyor.
}

```

```

class Nec {
    public:
        void set(int x, int y);
    private:
        int mx, my;
}
// nec.cpp
#include <nec.h>
Nec necgp;
void Nec::set(int x, int y)
{
    mx = x;
    my = x;
    // Class'ın kendi scopeunda private elemanlara erişebiliriz.
    necgp.mx = 123; // geçerli
}
// main.cpp
int main()
{
    Nec mynec;
    mynec.set(23, 12);
}

```

```

// Sınıfların üye fonksiyonları içinde isim arama (name lookup)
//nec.h
class Nec{
    public:
        void foo();
    private:
        int x;
}

//nec.cpp
int x = 45;
void Nec::foo()
{
    x = 45; // sınıf elemanı
    int x = 67;
    Nec::x = x + ::x
}

```

```

// member functions inline
// nec.h
class Nec{
    public:
        void set(int x, int y)
        {
            // implicitly inline
            mx = x;
            my = y;
        }
        inline int getx() const;
    private:
        int mx, my;
}
// header da tanımlayacaksak inline olarak ya bildirimde ya tanımda kullanmalıyız
inline int Nec::getx() const
{
    return mx;
};

```