



# BİL106 Nesne Yönelimli Programlama

Dr. Öğr. Üyesi Yavuz CANBAY  
Kahramanmaraş Sütçü İmam Üniversitesi  
Bilgisayar Mühendisliği Bölümü



---

# Bölüm 4: Yapılar

# Yapılar

---

- › Float, int, double, char vs gibi temel veri tipleri tek bir bilgiyi simgeler
- › Bir yapı ise, temel değişkenlerin bir araya getirilmesidir
- › Bir yapı içerisindeki değişkenler değişik tipte olabilir
  - › Örneğin int, float, char gibi
- › Bir yapı içerisindeki verilere yapının ÜYELERİ denir

# Basit bir YAPI örneği

```
// parts.cpp
// uses parts inventory to demonstrate structures
#include <iostream>
using namespace std;
////////////////////////////////////
struct part          //declare a structure
{
    int modelnumber;   //ID number of widget
    int partnumber;    //ID number of widget part
    float cost;        //cost of part
};
////////////////////////////////////
int main()
{
    part part1;        //define a structure variable

    part1.modelnumber = 6244; //give values to structure members
    part1.partnumber = 373;
    part1.cost = 217.55F;

    //display structure members
    cout << "Model "    << part1.modelnumber;
    cout << ", part "  << part1.partnumber;
    cout << ", costs $" << part1.cost << endl;
    return 0;
}
```

# Bir Yapının Sentaksı

```
struct part  
{  
    int modelnumber;  
    int partnumber;  
    float cost;  
};
```

Diagram illustrating the syntax of a structure definition:

- Keyword "struct"
- Structure name or "tag"
- Braces delimit structure members
- Structure members
- Semicolon terminates definition

# Yapı Üyelerine İlk Değer Ataması

```
// partinit.cpp
// shows initialization of structure variables
#include <iostream>
using namespace std;
////////////////////////////////////
struct part                //specify a structure
{
    int modelnumber;        //ID number of widget
    int partnumber;        //ID number of widget part
    float cost;            //cost of part
};
////////////////////////////////////
int main()
{
    //initialize variable
    part part1 = { 6244, 373, 217.55F };
    part part2;            //define variable
    //display first variable
    cout << "Model "      << part1.modelnumber;
    cout << ", part "    << part1.partnumber;
    cout << ", costs $"  << part1.cost << endl;

    part2 = part1;         //assign first variable to second
    //display second variable
    cout << "Model "      << part2.modelnumber;
    cout << ", part "    << part2.partnumber;
    cout << ", costs $"  << part2.cost << endl;
    return 0;
}
```

# Bir Ölçüm Örneği

```
// englstrc.cpp
// demonstrates structures using English measurements
#include <iostream>
using namespace std;
////////////////////////////////////
struct Distance          //English distance
{
    int feet;
    float inches;
};
////////////////////////////////////

int main()
{
    Distance d1, d3;          //define two lengths
    Distance d2 = { 11, 6.25 }; //define & initialize one length

                                //get length d1 from user
    cout << "\nEnter feet: "; cin >> d1.feet;
    cout << "Enter inches: "; cin >> d1.inches;

                                //add lengths d1 and d2 to get d3
    d3.inches = d1.inches + d2.inches; //add the inches
    d3.feet = 0;                      //(for possible carry)
    if(d3.inches >= 12.0)              //if total exceeds 12.0,
    {                                  //then decrease inches by 12.0
        d3.inches -= 12.0;            //and
        d3.feet++;                    //increase feet by 1
    }
    d3.feet += d1.feet + d2.feet; //add the feet

                                //display all lengths
    cout << d1.feet << "\'-" << d1.inches << "\" + ";
    cout << d2.feet << "\'-" << d2.inches << "\" = ";
    cout << d3.feet << "\'-" << d3.inches << "\"\n";
    return 0;
}
```

# İç İçe Yapılar

```
// englarea.cpp
// demonstrates nested structures
#include <iostream>
using namespace std;
////////////////////////////////////
struct Distance                //English distance
{
    int feet;

    float inches;
};
////////////////////////////////////
struct Room                    //rectangular area
{
    Distance length;           //length of rectangle
    Distance width;           //width of rectangle
};
////////////////////////////////////
int main()
{
    Room dining;               //define a room

    dining.length.feet = 13;    //assign values to room
    dining.length.inches = 6.5;
    dining.width.feet = 10;
    dining.width.inches = 0.0;

    //convert length & width
    float l = dining.length.feet + dining.length.inches/12;
    float w = dining.width.feet + dining.width.inches/12;
    //find area and display it
    cout << "Dining room area is " << l * w
         << " square feet\n" ;
    return 0;
}
```