

C++ Giriş Ders 10

MSGSU Fizik Bölümü

Ferhat ÖZOK

```
// example about structures
#include <iostream>
#include <string>
#include <sstream>
using namespace std;
struct movies_t {
    string title;
    int year;
} mine, yours;
void printmovie (movies_t movie);
int main ()
{
    string mystr;
    mine.title = "2001 A Space Odyssey";
    mine.year = 1968;
    cout << "Enter title: ";
    getline (cin,yours.title);
    cout << "Enter year: ";
    getline (cin,mystr);
    stringstream(mystr) >> yours.year;
    cout << "My favorite movie is:\n ";
    printmovie (mine);
    cout << "And yours is:\n ";
    printmovie (yours);
    return 0;
}
void printmovie (movies_t movie)
{
    cout << movie.title;
    cout << " (" << movie.year << ")\n"
```

Geçen dersimizde yaptığımız örnekte görüldüğü gibi structure objesinin üyeleri Normal değişkenlerdir. Örnek olarak yours objesinin üyesi yours.year geçerli int tipindedir

Ayrıca mine ve yours objeleri de bit tipi bulunan değişkenlerdir (movieas_t tipinde) printmovie fonksiyonuna parametre olarak basit tipi olan değişkenler gibi verilebiliyor. Data structure yapısının özelliklerinden biri üyelerine teker teker işaret edebileceği gibi yapının tamamını da işaret edebilir. Structurelar anı zamanda değişken tipi tanımladıkları için onları dizi tipi olarak kullanılabilirler böylelikle tablolar veritabanları structurelar vasıtasıyla oluşturulabilir.

C++ Giriş Ders 10

MSGSU Fizik Bölümü

Ferhat ÖZOK

```
// array of structures
#include <iostream>
#include <string>
#include <sstream>
using namespace std;
struct movies_t {
    string title;
    int year;
} films [3];
void printmovie (movies_t movie);
int main ()
{
    string mystr;
    int n;
    for (n=0; n<3; n++)
    {
        cout << "Enter title: ";
        getline (cin,films[n].title);
        cout << "Enter year: ";
        getline (cin,mystr);
        stringstream(mystr) >> films[n].year;
    }
    cout << "\nYou have entered these movies:\n";
    for (n=0; n<3; n++)
        printmovie (films[n]);
    return 0;
}
void printmovie (movies_t movie)
{
    cout << movie.title;
    cout << " (" << movie.year << ")\n";
}
```

Structure için işaretçiler

Diğer bütün değişken tipleri gibi structurelarda kendi tipindeki işaretçiler ile işaretlenebilirler

```
struct movies_t {  
    string title;  
    int year;  
};  
movies_t amovie;  
movies_t * pmovie;
```

amovie -> movies_t tipi structure tanımlayan obje

pmovie -> movies_t tipi structure objesini işaret eden bir işaretçi(pointer)

Aşağıdaki kullanımda geçerlidir

```
pmovie = &amovie;
```

C++ Giriş Ders 10

MSGSU Fizik Bölümü

Ferhat ÖZOK

```
// pointers to structures
#include <iostream>
#include <string>
#include <sstream>
using namespace std;
struct movies_t {
    string title;
    int year;
};
int main ()
{
    string mystr;
    movies_t amovie;
    movies_t * pmovie;
    pmovie = &amovie;
    cout << "Enter title: ";
    getline (cin, pmovie->title);
    cout << "Enter year: ";
    getline (cin, mystr);
    (stringstream) mystr >> pmovie->year;
    cout << "\nYou have entered:\n";
    cout << pmovie->title;
    cout << " (" << pmovie->year << ")\n";
    return 0;
}
```

Ok (arrow) operatörü (->) sadece üye elemanları bulunan objeler ile kullanılır. Bu operatör bir objenin elemanına direkt olarak adresinden ulaşılmasını sağlar.

pmovie->title == (*pmovie).title

Aşağıdaki kullanımdan farklıdır

*pmovie.title == *(pmovie.title)

C++ Giriş Ders 10

MSGSU Fizik Bölümü

Ferhat ÖZOK

İç içe girmiş yapılar(Nesting) structures

Bir structureun elamanı başka bir structure olabilir.

```
struct movies_t {  
    string title;  
    int year;  
};  
struct friends_t {  
    string name;  
    string email;  
    movies_t favorite_movie;  
} charlie, maria;  
friends_t * pfriends = &charlie;
```

Aşağıdaki kullanımların hepsi geçerlidir:

```
charlie.name  
maria.favorite_movie.title  
charlie.favorite_movie.year  
pfriends->favorite_movie.year
```

C++ Giriş Ders 10

MSGSU Fizik Bölümü

Ferhat ÖZOK

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

Örnek:

```
typedef struct telephone
{
    char *name;
    int number;
}TELEPHONE;
```

```
int main()
{
    TELEPHONE index;
    TELEPHONE *ptr_myindex;

    ptr_myindex = &index;

    ptr_myindex->name = "Jane Monroe";
    ptr_myindex->number = 12345;

    cout << "Name: " << ptr_myindex->name << '\n';
    cout << "Telephone number: " << ptr_myindex->number << '\n';

    return 0;
}
```

C++ Giriş Ders 10

MSGSU Fizik Bölümü

Ferhat ÖZOK

```
// This example shows the use of a structure in C++ and how it behaves much  
// like a class including the use of a constructor yet maintains the useability  
// of a regular C structure.
```

```
#include <iostream>
```

```
#include <string>
```

```
using namespace std;
```

```
main()
```

```
{
```

```
    struct DataElement {
```

```
        string SVal;
```

```
        int   iVal;
```

```
        bool  hasData;
```

```
        DataElement() // Example of a constructor used in a structure.
```

```
        {
```

```
            iVal=-1;
```

```
            hasData=0;
```

```
        }
```

```
    } *RealData;
```

```
RealData = new DataElement [ 5 ];
```

```
// Assignment
```

```
RealData[0].SVal = "Value loaded into first structure element.";
```

```
RealData[0].hasData = 1; // True
```

```
cout << "First element 0: " << RealData[0].SVal << endl;
```

```
cout << "                " << RealData[0].hasData << endl;
```

```
cout << "Second element 1: " << RealData[1].SVal << endl;
```

```
cout << "                " << RealData[1].hasData << endl; // Show effect of constructor
```

```
cout << "                " << RealData[1].iVal  << endl; // Show effect of constructor
```

```
delete [] RealData; // Or: delete [5] RealData;
```

```
}
```

Örnek:

C++ Giriş Ders 10

MSGSU Fizik Bölümü

Ferhat ÖZOK

Unions

Unions belli bir hafıza birimine farklı data tipleri olarak ulaşımaya imkan verir. Yapısı ve tanımlanması Structure tanımlaması ile benzerdir ancak kullanımı tamamen farklıdır.

Yapısı:

```
union ismi {  
    member_type1 member_name1;  
    member_type2 member_name2;  
    member_type3 member_name3;  
    .  
    .  
} object_ismi;
```

Bu şekilde ismi ile tanımlanan bir union tipi oluşturulur. Bu yapıda bütün elemanlar aynı fiziksel hafıza alanını işgal eder.

Bu tipin boyutu en büyük elemanının kapladığı alan kadardır.

Örnek:

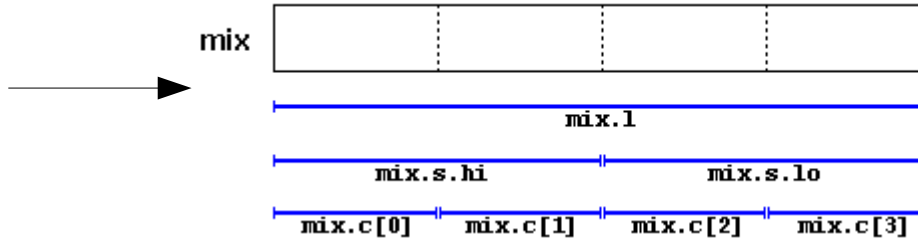
```
union mytypes_t {  
    char c;  
    int i;  
    float f;  
} mytypes;
```


C++ Giriş Ders 10

MSGSU Fizik Bölümü

Ferhat ÖZOK

```
union mix_t {
    int l;
    struct {
        short hi;
        short lo;
    } s;
    char c[4];
} mix;
```



structure with regular union	structure with anonymous union
<pre>struct book1_t { char title[50]; char author[50]; union { float dollars; int yen; } price; } book1;</pre>	<pre>struct book2_t { char title[50]; char author[50]; union { float dollars; int yen; }; } book2;</pre>

book1.price.dollars
book1.price.yen

book2.dollars
book2.yen

Enumerated types (enum)

Bu tip, değişkenin alabileceği değerlerin belli (sabit) olduğu durumlarda programı daha okunabilir hale getirmek için kullanılır. Genel yazım biçimi:

```
enum tip_adi{değer_1, değer_2, ..., değer_n} değişken_adi;  
tip_adi programcı tarafından verilen tip ismidir. değişken_adi ise program içinde  
kullanılacak olan değişkenin adıdır. Eğer kullanılmazsa program içinde daha  
sonra enum ile birlikte kullanılır. Örneğin:  
enum bolumler{fizik, matematik, kimya, elektrik};  
tanımı ile derleyici programcılık için 0, donanim için 1, muhasebe için 2  
ve motor için 3 değerini kabul ederek atamaları buna göre yapar. Değişken adı  
bildirilirse daha sonra enum kullanmaya gerek kalmaz. Örneğin:  
enum renkler {kirmizi, mavi, sari} renk;  
enum gunler {pazartesi, sali, carsamba, persembe, cuma, cumartesi, pazar};  
gibi yapılan sabit tanımlamaları program içinde kullanılabilir:  
enum bolumler bolum;  
enum gunler gun;  
...  
bolum = kimya; /* bolum = 2 anlamında */  
gun = persembe; /* gun = 3 anlamında */  
renk = kirmizi; /* renk = 0 anlamında */
```

C++ Giriş Ders 10

MSGSU Fizik Bölümü

Ferhat ÖZOK

ÖRNEKLER

C++ Giriş Ders 10

MSGSU Fizik Bölümü

Ferhat ÖZOK

```
#include<iostream>
using namespace std;
int main ()
{
    int num[5];
    int * ptr_p;
    ptr_p = num; //ptr_p points at first element
    *ptr_p = 1; //Store the value 1
    ptr_p++; //Increase to second element
    *ptr_p = 2; //Store the value 2
    ptr_p = &num[2]; //Get adres of third
element
    *ptr_p = 3; //Store the value 3
    ptr_p = num + 3; //Goto element 4
    *ptr_p = 4; //Store the value 4
    ptr_p = num; //Point at first element
store 5
    *(ptr_p+4) = 5; //First goto element 5 and then

    //Now print value of each element
    for (int i=0; i<5; i++)
        cout << num[i] << '\n';
    return 0;
}
```

C++ Giriş Ders 10

MSGSU Fizik Bölümü

Ferhat ÖZOK

```
// yinelemeli fonksiyon
#include <cmath> //
#include <iostream> //

using namespace std;
long faktoriyal (long a)
{
    if (a > 1){
        cout << a<<endl;
        return (a * faktoriyal (a-1));
    }else
        return (1);
}
int main ()
{
    long sayi;
    do{
        cout << "Bir sayi giriniz: ";
        cin >> sayi;
        cout << sayi << "! = " << faktoriyal (sayi);
    }while (sayi >= 32);

    return 0;

}
```

C++ Giriş Ders 10

MSGSU Fizik Bölümü

Ferhat ÖZOK

```
include <cmath> //
#include <iostream> //
using namespace std;
long fakt(int n)
{
    if (n < 0) return 0;
    int f = 1;
    while (n > 1) f *= n--;
    return f;
}

long perm(int n,int k)
{
    if (n < 0 || k < 0 || k > n) return 0;
    return fakt(n)/fakt(n-k);
}

int main()
{
    for (int i = -1; i < 8; i++)
    { for (int j=-1; j <= i+1; j++)
        cout << " " << perm(i,j);
        cout << endl;
    }
    return 0;
}
```

```
#include <stdio.h>
struct ogrenci
{
    char isim[50];
    int numarasi;
    float notu;
} s;
int main()
{
    printf("Bilgileri giriniz:\n");
    printf("isim giriniz: ");
    scanf("%s", s.isim);
    printf("Ogrenci no giriniz: ");
    scanf("%d", &s.numarasi);
    printf("Notunu giriniz: ");
    scanf("%f", &s.notu);
    printf("Bilgileri Goster:\n");
    printf("İsim: ");
    puts(s.isim);
    printf("Ogrenci numarasi: %d\n", s.numarasi);
    printf("Notu: %.1f\n", s.notu);
    return 0;
}
```

C++ Giriş Ders 10

MSGSU Fizik Bölümü

Ferhat ÖZOK

```
#include<iostream>
using namespace std;
int main ()
{
    int num[5];
    int * ptr_p;
    ptr_p = num; //ptr_p points at first element
    *ptr_p = 1; //Store the value 1
    ptr_p++; //Increase to second element
    *ptr_p = 2; //Store the value 2
    ptr_p = &num[2]; //Get adres of third element
    *ptr_p = 3; //Store the value 3
    ptr_p = num + 3; //Goto element 4
    *ptr_p = 4; //Store the value 4
    ptr_p = num; //Point at first element
    *(ptr_p+4) = 5; //First goto element 5 and then store 5
    //Now print value of each element
    for (int i=0; i<5; i++)
        cout << num[i] << '\n';
    return 0;
}
```


C++ Giriş Ders 10

MSGSU Fizik Bölümü

Ferhat ÖZOK

```
#include <stdio.h>

void merge(int [], int, int [], int, int []);

int main() {
    int a[100], b[100], m, n, c, sorted[200];

    printf("ilk dizinin eleman sayisini giriniz\n");
    scanf("%d", &m);

    printf("Input %d tam sayi\n", m);
    for (c = 0; c < m; c++) {
        scanf("%d", &a[c]);
    }

    printf("ikinci dizinin eleman sayisini giriniz\n");
    scanf("%d", &n);

    printf("Input %d Tam sayi\n", n);
    for (c = 0; c < n; c++) {
        scanf("%d", &b[c]);
    }

    merge(a, m, b, n, sorted);

    printf("Siralanmis yeni dizi:\n");

    for (c = 0; c < m + n; c++) {
        printf("%d\n", sorted[c]);
    }

    return 0;
}
```

```
void merge(int a[], int m, int b[], int n, int sorted[]) {
    int i, j, k;

    j = k = 0;

    for (i = 0; i < m + n; i) {
        if (j < m && k < n) {
            if (a[j] < b[k]) {
                sorted[i] = a[j];
                j++;
            }
            else {
                sorted[i] = b[k];
                k++;
            }
            i++;
        }
        else if (j == m) {
            for (; i < m + n; i) {
                sorted[i] = b[k];
                k++;
                i++;
            }
        }
        else {
            for (; i < m + n; i) {
                sorted[i] = a[j];
                j++;
                i++;
            }
        }
    }
}
```