C++ Giriş Ders 7
MSGSU Fizik Bölümü
Ferhat ÖZOK

Daha önceki derslerimizde gördüğümüz gibi birden farkli türde aynı isimde fonksion tanımlayabiliriz. Bunların hepsini tekbiryapıda toplamak daha mantıklı olabilir. Bu tip durumlar için C++'ta bunun için generik tipte fonksiyon tanımlana bilmektedir. Bu tip fonksiyonlara şablon(tamplate) fonksiyon denir. Yapısı:

template <template-parameters> function-declaration

Bir şablon fonksiyon için örnek kalıp olara iki sayının toplamını alan bir şablon fonksiyonu düşünelim bu şablon fonksiyonun yapısı

```
template <class SomeType>

SomeType sum (SomeType a, SomeType b)

{

  return a+b;

}
```

Şeklinde olacaktır.

MİMAR SİNAN FINE ARTS UNIVERSITY

C++ Giriş Ders 7
MSGSU Fizik Bölümü
Ferhat ÖZOK

Örnek 1:

```
// function template
#include <iostream>
using namespace std;
template <class T>
T sum (T a, T b)
{
  T result;
  result = a + b;
  return result;
}
int main () {
  int i=5, j=6, k;
  double f=2.0, g=0.5, h;
  k=sum<int>(i,j);
  h=sum<double>(f,g);
  cout << k << '\n';
  cout << h << '\n';
  return 0;
}
```

C++ Giriş Ders 7
MSGSU Fizik Bölümü
Ferhat ÖZOK

Örnek 2:

Şablonlarda parametreler sadece type yada class name ile değil bazen belirli bir tip içinde tanımlanabilir

```
// template arguments
#include <iostream>
using namespace std;
template <class T, int N>
T fixed_multiply (T val)
{
  return val * N;
}
int main() {
  std::cout << fixed_multiply<int,2>(10) <<
'\n';
  std::cout << fixed_multiply<int,3>(10) <<
'\n';
}
```

C++ Giriş Ders 7
MSGSU Fizik Bölümü
Ferhat ÖZOK

Genel(Global) Özel (Lokal) Değişkenler

```
int foo;        // global değişken
int some_function ()
{
  int bar;      // yerel variable
  bar = 0;
}
int other_function ()
{
  foo = 1;  // ok: foo is a global değişken
  bar = 2;  // yanlış: bar bu fonksiyon için görünür değil
}
```

```
int some_function ()
{
  int x;
  x = 0;
  double x;   // yanlış : bu isimde bir değişken önceden tanımlandı
x = 0.0;
}
```

Örnek

```
// inner block scopes
#include <iostream>
using namespace std;
int main () {
  int x = 10;
  int y = 20;
  {
    int x;   // ok, inner scope.
    x = 50;  // sets value to inner x
    y = 50;  // sets value to (outer) y
    cout << "inner block:\n";
    cout << "x: " << x << '\n';
    cout << "y: " << y << '\n';
  }
  cout << "outer block:\n";
  cout << "x: " << x << '\n';
  cout << "y: " << y << '\n';
  return 0;
}
```

C++ Giriş Ders 7
MSGSU Fizik Bölümü
Ferhat ÖZOK

Namespaces:

Kullanım şekli:

namespace tanımlayıcı

{

Isimlendirilmş elemanlar

}

Örnek:

```
namespace
myNamespace
{
  int a, b;
}
```

Örnek 1:

```
// namespaces
#include <iostream>
using namespace std;
namespace foo
{
  int value() { return 5; }
}
namespace bar
{
  const double pi = 3.1416;
  double value() { return 2*pi; }
}
int main () {
  cout << foo::value() << '\n';
  cout << bar::value() << '\n';
  cout << bar::pi << '\n';
  return 0;
}
```

C++ Giriş Ders 7
MSGSU Fizik Bölümü
Ferhat ÖZOK

Örnek 2:

```
// using
#include <iostream>
using namespace std;
namespace first
{
  int x = 5;
  int y = 10;
}
namespace second
{
  double x = 3.1416;
  double y = 2.7183;
}
int main () {
  using first::x;
  using second::y;
  cout << x << '\n';
  cout << y << '\n';
  cout << first::y << '\n';
  cout << second::x << '\n';
  return 0;
}
```

MİMAR SİNAN FINE ARTS UNIVERSITY

C++ Giriş Ders 7
MSGSU Fizik Bölümü
Ferhat ÖZOK

Örnek 3:

```
// using
#include <iostream>
using namespace std;
namespace first
{
  int x = 5;
  int y = 10;
}
namespace second
{
  double x = 3.1416;
  double y = 2.7183;
}
int main () {
  using namespace first;
  cout << x << '\n';
  cout << y << '\n';
  cout << second::x << '\n';
  cout << second::y << '\n';
  return 0;
}
```

C++ Giriş Ders 7
MSGSU Fizik Bölümü
Ferhat ÖZOK

Her değişken için bilgisayar hafızasında bir yer ayrılır. Global değişkenler ve namespaceler için program süresince yer ayrılır. Bu na statik hafıza alanı denir. Yerel değişkenler için ise ayrılan yer sadece programda o değişkenin tanımlandığı blokta oluşturulur. Bu alanlara da dinamik hafızaalanı olarak tanımlanır

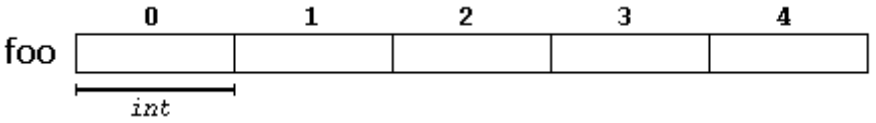
- Global değişkenler ve namespaceler eğer ildeğer verilmez ise otomatik olarak ilk değer olarak 0 atanır.
- Yerel değişkenler ise ilk değerlenidirme otomatik olarak yapılmaz kullanıcı tarafından ilk değer verilmelidir

Örnek:

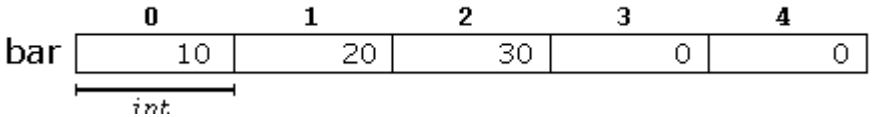
```
// static vs automatic storage
#include <iostream>
using namespace std;
int x;
int main ()
{
  int y;
  cout << x << '\n';
  cout << y << '\n';
  return 0;
}
```

C++ Giriş Ders 7
MSGSU Fizik Bölümü
Ferhat ÖZOK

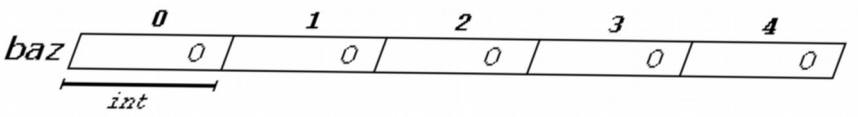
Diziler: Dizi ayni tür değişken serisininsurekli bir hafıza alanında tutulmasıdır. C++ da dizi yapısı:
type isim [eleman sayısı];
Ör: 5 elemanlı integer dizi
int foo [5];




```
int foo [5] = { 16, 2, 77, 40, 12071 };
```

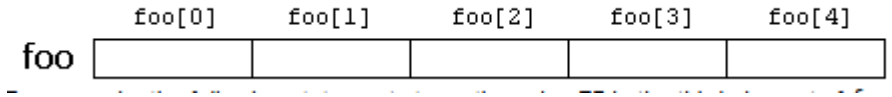



```
int baz [5] = { };
```



C++ Giriş Ders 7
MSGSU Fizik Bölümü
Ferhat ÖZOK

Dizinin bir elemanına erişmek:




```
foo [2] = 75;
(Foo nun 3. elemanı)
x = foo[2];
```

```
foo[0] = a;
foo[a] = 75;
b = foo [a+2];
foo[foo[a]] = foo[2] + 5;
```

Örnek:

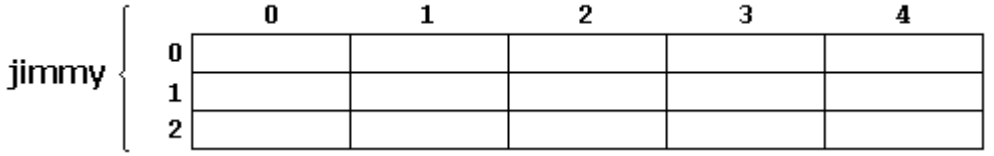
```
// arrays example
#include <iostream>
using namespace std;
int foo [] = {16, 2, 77, 40, 12071};
int n, result=0;
int main ()
{
  for ( n=0 ; n<5 ; ++n )
  {
    result += foo[n];
  }
  cout << result;
  return 0;
}
```

MİMAR SİNAN FINE ARTS UNIVERSITY

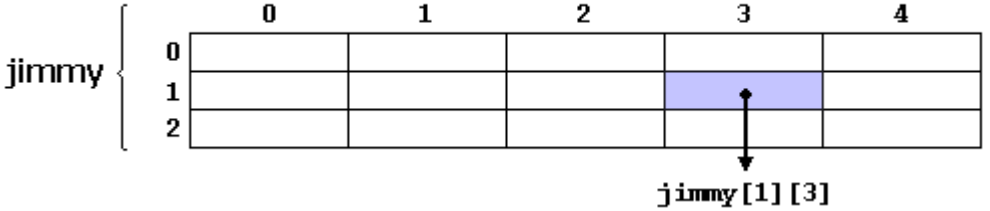
C++ Giriş Ders 7
MSGSU Fizik Bölümü
Ferhat ÖZOK

Çok Boyutlu Diziler

```
int jimmy [3][5];
```




```
jimmy[1][3]
```




```
#define WIDTH 5
#define HEIGHT 3

int jimmy [HEIGHT][WIDTH];
int n,m;

int main ()
{
  for (n=0; n<HEIGHT; n++)
    for (m=0; m<WIDTH; m++)
    {
      jimmy[n][m]=(n+1)*(m+1);
    }
}
```

```
#define WIDTH 5
#define HEIGHT 3

int jimmy [HEIGHT * WIDTH];
int n,m;

int main ()
{
  for (n=0; n<HEIGHT; n++)
    for (m=0; m<WIDTH; m++)
    {
      jimmy[n*WIDTH+m]=(n+1)*(m+1);
    }
}
```

jimmy

	0	1	2	3	4
0	1	2	3	4	5
1	2	4	6	8	10
2	3	6	9	12	15

C++ Giriş Ders 7
MSGSU Fizik Bölümü
Ferhat ÖZOK

Parametre olarak diziler

```
// arrays as parameters
#include <iostream>
using namespace std;
void printarray (int arg[], int length)
{
  for (int n=0; n<length; ++n)
    cout << arg[n] << ' ';
  cout << '\n';
}
int main ()
{
  int firstarray[] = {5, 10, 15};
  int secondarray[] = {2, 4, 6, 8, 10};
  printarray (firstarray,3);
  printarray (secondarray,5);
}
```

(int arg[]) bu parametre bilesenleri int olan istenilen uzunlukta olabilecek dizidir

```
   void procedure (int myarray[][3][4])
```

MİMAR SİNAN FINE ARTS UNIVERSITY

C++ Giriş Ders 7
MSGSU Fizik Bölümü
Ferhat ÖZOK

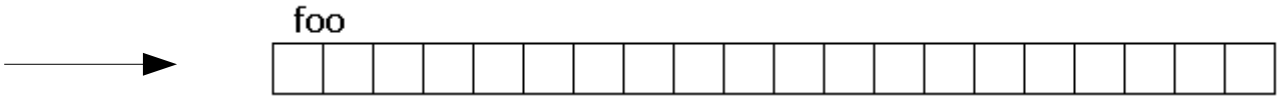
Kütüphane dizileri
Alrternatif olarak C++ <array> standard header ile alternatif bir dizi kullanımı sağlar
Bu dizi aslında bir şablon fonksiyondur

language built-in array	container library array

`#include <iostream>`
 	
`using namespace std;`
 	
`int main()`
`{`
`  int myarray[3] = {10,20,30};`
 	
`  for (int i=0; i<3; ++i)`
`  ++myarray[i];`
 	
`  for (int elem : myarray)`
`  cout << elem << '\n';`
`}`
	`#include <iostream>`
	`#include <array>`
	`using namespace std;`
	
	`int main()`
	`{`
	`  array<int,3> myarray {10,20,30};`
	
	`  for (int i=0; i<myarray.size(); ++i)`
	`  ++myarray[i];`
	
	`  for (int elem : myarray)`
	`  cout << elem << '\n';`
	`}`

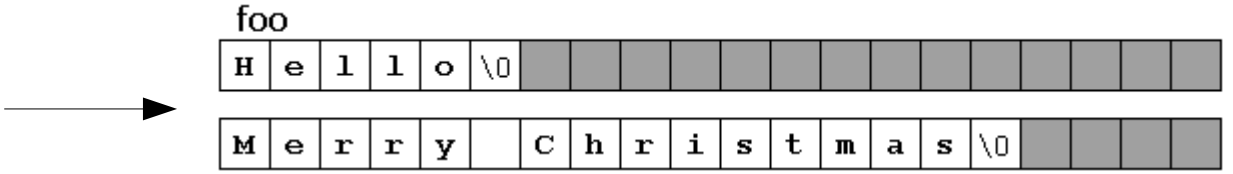
C++ Giriş Ders 7
MSGSU Fizik Bölümü
Ferhat ÖZOK

```
char foo [20];
```



Değeri

"Hello" ve "Merry Christmas"



`\0` ifadesi karakter katarının sonuna gelindiğini gösteriri

```
char myword[] = { 'H', 'e', 'l', 'l', 'o', '\0' };
char myword[] = "Hello";
```

Diziler üste tanımlandıktan sonra yeni değer atamak istenirse

```
myword = "Bye";
myword[] = "Bye";
myword = { 'B', 'y', 'e',
'\0' };
```

YANLIŞ!!!

```
myword[0] = 'B';
myword[1] = 'y';
myword[2] = 'e';
myword[3] = '\0';
```

DOĞRU(GEÇERLİ)!!

MİMAR SİNAN FINE ARTS UNIVERSITY

C++ Giriş Ders 7
MSGSU Fizik Bölümü
Ferhat ÖZOK

```
// strings and NTCS:
#include <iostream>
#include <string>
using namespace std;
int main ()
{
  char question1[] = "What is your name? ";
  string question2 = "Where do you live? ";
  char answer1 [80];
  string answer2;
  cout << question1;
  cin >> answer1;
  cout << question2;
  cin >> answer2;
  cout << "Hello, " << answer1;
  cout << " from " << answer2 << "!\n";

char myntcs[] = "some text";
string mystring = myntcs;  // convert c-string to string
cout << mystring;          // printed as a library string
cout << mystring.c_str();  // printed as a c-string

  return 0;
}
```