C++ Giriş Ders 6
MSGSU Fizik Bölümü
Ferhat ÖZOK

Fonksiyonlar:
Belli işlemleri yapmaya izin veren program yapısıdır. C++ da bir grup işlemlerin belli bir isimde tanımlanması ve programın istendiği yerde çağrılabilir. Fonksiyon sintaksi:

Tür isim (parametre1,parametre2,parametre3,…) { işlemler }

Burda
Tür: fonksiyonlar tarafından yapılacak işlem(ler) sonucunda döndüreceği sonucun türünü belirtir.
İsim: Fonksiyonun çağrılacağı onu tanımlayan ismini belirtir.
Parametreler (gerektiği kadar olabilir): Her parametre belirlenen tipte tanımlıyıcıya sahiptir. Her parametre “,” ile ayrılır ve normal değişkenler giibi tanımlanır.
Işlemler {} arasında belirlenir.

MİMAR SİNAN FINE ARTS UNIVERSITY

C++ Giriş Ders 6
MSGSU Fizik Bölümü
Ferhat ÖZOK

Örnek 1:

```
// fonksiyon  örneği
#include <iostream>
using namespace std;
int Toplama (int a, int b)
{
  int r;
  r=a+b;
  return r;
}
int main ()
{
  int z;
  z = Toplama (5,3);
  cout << "The result is " << z;
}
```

C++ Giriş Ders 6
MSGSU Fizik Bölümü
Ferhat ÖZOK

Örnek 2:

```
// function örnek
#include <iostream>
using namespace std;
int Cikartma (int a, int b)
{
  int r;
  r=a-b;
  return r;
}
int main ()
{
  int x=5, y=3, z;
  z = Cikartma (7,2);
  cout << "Ilk sonuc: " << z << '\n';
  cout << "Ikinci sonuc: " << Cikartma (7,2) << '\n';
  cout << "ucuncu sonuc: " << Cikartma (x,y) << '\n';
  z= 4 + Cikartma (x,y);
  cout << "dorduncu sonuc: " << z << '\n';
}
```

MİMAR SİNAN FINE ARTS UNIVERSITY

C++ Giriş Ders 6
MSGSU Fizik Bölümü
Ferhat ÖZOK

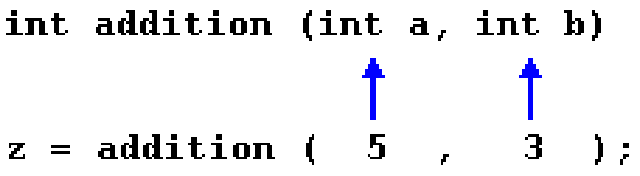
Tipi olmayan Fonksiyonlar:
Void fonksiyon ismi (parametre1,parametre2,...){işlemler}

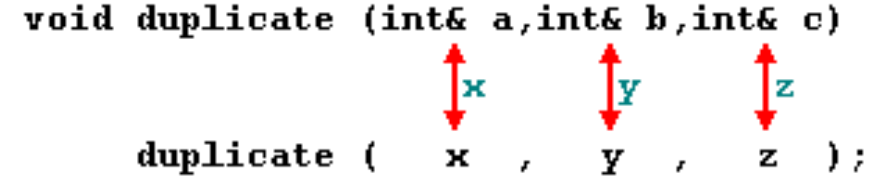
Örnek 3:

```
// void function example
#include <iostream>
using namespace std;
void printMesaji ()
{
  cout << "Ben bir Fonksiyonum!";
}
int main ()
{
  printMesaji ();
}
```

C++ Giriş Ders 6
MSGSU Fizik Bölümü
Ferhat ÖZOK

Argumanların değer yada referans ile geçirilmesi





Örnek 4:

```
// passing parameters by reference
#include <iostream>
using namespace std;
void ikiyekatla (int& a, int& b, int& c)
{
  a*=2;
  b*=2;
  c*=2;
}
int main ()
{
  int x=1, y=3, z=7;
  ikiyekatla (x, y, z);
  cout << "x=" << x << ", y=" << y << ", z=" << z;
  return 0;
}
```

C++ Giriş Ders 6
MSGSU Fizik Bölümü
Ferhat ÖZOK

Örnek 5:

```
// default values in functions
#include <iostream>
using namespace std;
int Bolen (int a, int b=2)
{
  int r;
  r=a/b;
  return (r);
}
int main ()
{
  cout << Bolen (12) << '\n';
  cout << Bolen (20,4) << '\n';
  return 0;
}
```

MİMAR SİNAN FINE ARTS UNIVERSITY

C++ Giriş Ders 6
MSGSU Fizik Bölümü
Ferhat ÖZOK

Fonksiyonların önceden tanımlanması

Örnek 6:

```
// declaring functions prototypes
#include <iostream>
using namespace std;
void Tek (int x);
void Cift (int x);
int main()
{
  int i;
  do {
    cout << "Lütfen bir sayı giriniz (cikmak icin 0 giriniz): ";
    cin >> i;
    Tek (i);
  } while (i!=0);
  return 0;
}
void Tek (int x)
{
  if ((x%2)!=0) cout << "Sayi Tek.\n";
  else Cift (x);
}
void Cift (int x)
{
  if ((x%2)==0) cout << "Sayi Çift.\n";
  else Tek (x);
}
```

MİMAR SİNAN FINE ARTS UNIVERSITY

C++ Giriş Ders 6
MSGSU Fizik Bölümü
Ferhat ÖZOK

Recursivity (Kendi kendini tekrarlama)

```
// factorial calculator
#include <iostream>
using namespace std;
long factorial (long a)
{
  if (a > 1)
   return (a * factorial (a-1));
  else
   return 1;
}
int main ()
{
  long number = 9;
  cout << number << "! = " << factorial (number);
  return 0;
}
```

C++ Giriş Ders 6
MSGSU Fizik Bölümü
Ferhat ÖZOK

Aynı isimle birden fazla Fonksiyonun tanımlanması (overloading functions)

```
// overloading functions
#include <iostream>
using namespace std;
int Hesapla (int a, int b)
{
  return (a*b);
}
double Hesapla (double a, double b)
{
  return (a/b);
}
int main ()
{
  int x=5,y=2;
  double n=5.0,m=2.0;
  cout << Hesapla (x,y) << '\n';
  cout << Hesapla (n,m) << '\n';
  return 0;
}
```