**C ile Programlama Doküman 3**

**(Fonksiyonlar)**

C ile programlamadaki alt programlara fonksiyon denilmektedir. Örneğin Java alt programlara metot denilmektedir. C++ dilinde C’de olduğu gibi fonksiyon ismi kullanılmaktadır. Bir fonksiyon ana programdan bağımsız çalışabilen bir program parçasıdır. Bir programı fonksiyonlara bölerek yazmak bazı faydalar sağlar:

-Programın kaynak kodu küçülür. Böylece byte kod da küçülür.

-Kaynak kodun okunabilirliği ve anlaşılabilirliği artar.

-Belirli kod parçalarının programın farklı yerlerinde yinelenmesi, programda yapılacak olası bir değişikliğin maliyetini azaltır. Ortak kod parçaları fonksiyonlar olarak yazıldığında yalnızca fonksiyonlarda yapılan değişiklikler yeterlidir.

-Programda bir hata olduğunda code review yapmak kolaylaşır.

-Yazılan fonksiyonlar başka projelerde de kullanılabilir.

C dilinde fonksiyon içerisinde fonksiyon bildirimi geçersizdir.

**Fonksiyonların Geri Dönüş Değerleri**

Bir fonksiyon çağrıldığı zaman onun kodu çalıştırılır. Fonksiyon sona erdiğinde ismine geri dönüş değeri (return value) denilen bir değer elde edilir. Bu değer ifadelerde kullanılabilir. Fonksiyonların geri dönüş değerlerinin türü fonksiyon isminin soluna yazılır.

Örneğin:

int foo(void)

{

//fonksiyon gövdesi

}

Fonksiyonların geri dönüş değerleri return deyimiyle oluşturulur. return deyimiyle genel biçimi şöyledir:

return [ifade(expression)];

return deyiminin iki tane işlevi vardır:

1. Geri dönüş değerini oluşturur.
2. Fonksiyonu sonlandırır. Yani onun aşağısına yazacağımız kod çalıştırılmaz. Diğer bir değişle unreachable code olur.

Fonksiyonun geri dönüş değeri varsa kesinlikle return işlemi uygulanması zorunludur. Bu bazı dillerde olduğu gibi syntax hatası olmasa da yazılması asla tavsiye edilmez. Örneğin:

#include <stdio.h>

int foo(void)//Undefined behaviour

{

}

int main()

{

printf(“%d”,foo());

}

Ek olarak fonksiyonun her mümkün akışında da return anahtar sözcüğü görülmesi gerekir. Geri dönüş değeri türünün yerine void yazılırsa bu durum fonksiyonun geri dönüş değeri olmadığı anlamına gelir. Geri dönüş değeri olmayan fonksiyonlar geri dönüş değeri varmış gibi kullanılamaz.

#include <stdio.h>

void foo()

{

printf(“foo called\n”);

}

int main()

{

int result;

result = foo();

}

void bir fonksiyonda return deyimi tek başına kullanılabilir. Bu bir zorunluluk değildir. Void metotlarda return deyimi kullanılırsa return deyiminin yanına bir ifade yazılamaz. Bu durumda return fonksiyonu sonlandırmak için kullanılır. void fonksiyonlarda return kullanılmışsa fonksiyon ana blok bittiğinde biter. Bir fonksiyonun geri dönüş değerinin olması onu kullanmayı zorunlu hale getirmez. Yani geri dönüş değerleri programcı tarafından kullanılmayabilir.

Fonksiyonların geri dönüş değerleri önce derleyici tarafından bir geçici değişkene aktarılır. Oradan alınarak kullanılır. İfade sonunda o geri dönüş değeri yok edilir. Örneğin:

a = foo() \* 10;

ifadesinde aslında şunlar olmaktadır.

temp = return ifadesi;

a = temp \* 10;

temp yok ediliyor.

Yani aslında return işlemi geçici değişkene yapılan bir atama işlemidir. Metotların geri dönüş değerlerinin türü aslında yaratılacak olan geçici değişkenin türünü belirler.

**Fonksiyonların Parametre Değişkenleri**

Fonksiyonların parametre değişkenleri fonksiyonun girdileri olarak düşünülebilir. Fonksiyonların parametre değişkenleri fonksiyon bildiriminde parametre parantezi içerisinde belirtilir. Parametre değişkenleri fonksiyon bildirimi sırasında tür ve değişken ismi virgül atomu ile ayrılacak şekilde yapılır:

void foo(int a, double b)

{

//...

}

Parametre değişkenleri aynı türden olsa bile tür belirten sözcük her parametre için yazılmalıdır.

void foo(int a, b)//error

{

//...

}

Bu metodun şu şekilde bildirilmesi gerekir:

void foo(int a, int b)

{

//...

}

**Anahtar Notlar:** C programlama dilinde (hatta birçok programlama dilinde) fonksiyon/metot bildiriminde kullanılan değişkenlere parametre değişkenler, fonksiyonun çağrılması sırasında parantez içerisinde verilen ifadelere de argüman denilmektedir.

Parametreli bir fonksiyon parametre sayısı kadar argümanla çağrılmak zorundadır. Argümanlar herhangi bir ifade olabilir. Örneğin:

foo(10 + 5,11 \* 13);

Bir fonksiyona argüman olarak başka bir fonksiyonun geri dönüş değeri verilebilir. Bu durumda önce argüman olarak verilen fonksiyon çağrılır, daha sonra fonksiyonun geri dönüş değeri argüman olarak aktarılır. Örneğin:

#include <stdio.h>

int square(int val)

{

return val \* val;

}

void foo(int a,double b)

{

printf(“a : %d\n”,a);

printf(“b : %d\n”,b);

}

int main()

{

int a = 10;

double pi = 3.14159;

foo(square(a),pi \* pi);

}

Parametreli bir fonksiyon çağrıldığında önce argümanların değerleri hesaplanır. Sonra argümanlardan parametre değişkenlerine otomatik atama yapılır. Yani fonksiyon çağrısı sırasında verilen argümanlar da parametrelere yapılan atama işlemidir. Sonra da akış fonksiyona geçer.

**Anahtar Notlar:** Bir fonksiyon eğer parametre değişkenine sahip değilse o fonksiyonun parametre parantezinin içine void yazılmalıdır. Bunun nedenini ilerde ele alacağız.

Örneğin aşağıdaki örnekte foo fonksiyonu istenildiği kadar parametreyle çağrılabilir.Ancak C++ dilinde böyle bir durum söz konusu değildir.

int foo()

{

}

int main()

{

foo(10);//geçerli

foo(34,56,78);//geçerli

foo(15,20,30,67,89);//geçerli

}

**Anahtar Notlar:** C programlama dilinde aşağıdaki durumlar aslında atama işlemleridir:

-Atama operatörü ile yapılan atamalar

-Bir fonksiyonun geri dönüş değeri: Geçici değişkene yapılan atama

-Parametreli bir fonksiyonun çağrılması: Argümanlardan parametrelere yapılan atama

Fonksiyonların parametre değişkenleri yalnızca o metot içerisinde kullanılabilir. Fonksiyonların parametre değişkenleri adeta fonksiyonların ana bloğunun başında bildirilmiş yerel değişkenler gibidir. Dolayısıyla fonksiyon içerisinde bir parametre değişkeni ile aynı isimli başka bir yerel değişken bildirilemez. Ek olarak parametre değişkenlerini fonksiyon içerisinde kullanılmaması durumu syntax hatasına neden olmaz ancak kullanılmadığı sürece anlamlı değildir.

**C’de Matematiksel Fonksiyonlar**

C dilinde matematiksel işlemlerin yapılması için standart kütüphanenin bazı fonksiyonları bulunmaktadır. Örneğin karekök alma sinüs alma vb. Bu fonksiyonlar programda doğrudan kullanılamaz. Bu fonksiyonların kullanımı için math.h başlık dosyasının programa eklenmesi gerekmektedir. Bu fonksiyonların programcı tarafından yazılması da elbette mümkündür ancak bunun için belli düzeyde bir matematik bilgisi gerekmektedir. Programcı bu fonksiyonları hazır kullanırken bu detayları bilmek zorunda değildir. Çok kullanılan bazı standart kütüphane fonksiyonları aşağıdaki gibidir:

-math.h başlık dosyasının sqrt fonksiyonu parametre olarak aldığı sayının karekökünü geri döner.

-math.h başlık dosyasının abs fonksiyonu parametre olarak aldığı sayının mutlak değerini geri döner.

-math.h başlık dosyasının sin, cos, tan, asin, acos, atan gibi fonksiyonları trigonometrik işlemler yapar. Bu fonksiyonların parametre ya da geri dönüş değerleri radyan cinsindendir.

-math.h başlık dosyasının pow fonksiyonu kuvvet alma işlemi yapar.

-math.h başlık dosyasının exp fonksiyonu parametre olarak aldığı değeri kuvvet kabul ederek e sayısının o kuvvetini geri döner.

-math.h başlık dosyasının signum fonksiyonu parametre olarak aldığı değerin işaretini belirleyen fonksiyondur. Sayı negatifse -1, pozitifse 1 ve sıfırsa 0(sıfır) değerini geri döner.

- math.h başlık dosyasının log, log10 ve log1p metotları sırasıyla e tabanına göre logaritma [ln(val)], 10 tabanına göre logaritma ve parametre olarak aldığı değerin 1(bir) fazlasının e tabanına göre logaritma değerlerine [ln(val + 1)] geri dönerler. Herhangi bir tabana göre logaritma işlemi için programcı bu metotları kullanabilir.