

5..Ders : C Giriş / Çıkış

Bu derste, C programlamada I/O görevi gerçekleştirmek için iki yerleşik işlev printf () ve scanf () üzerinde durulur. Ayrıca, C de geçerli bir program yazmayı öğreneceksiniz.

C programlama, giriş ve çıkış görevlerini gerçekleştirmek için çeşitli yerleşik kitaplık işlevlerine sahiptir.

Input Output (I/O) [(Giriş / Çıkış)] için yaygın olarak kullanılan iki işlev printf () ve scanf () 'dir.

Scanf () işlevi, standart girdiden (klavye) biçimlendirilmiş girişi okur, printf () işlevi standart çıktıya (ekrana) biçimlendirilmiş çıktı gönderir.

▪ Örnek 1 : C çıkış (C Output)

```
#include <stdio.h> // Printf () fonksiyonunu çalıştırmak için
başlık dosyasını dahil ediyoruz
int main()
{
    printf("C Programlama");    // geçerli içeriği
    görüntülüyoruz
    return 0;
}
```

Çıktısı :

C Programlama

Bu program nasıl çalışır?

Tüm geçerli C programları main () işlevini içermelidir. Kod yürütme, main () fonksiyonunun başlangıcından başlar.

Printf (), formatlanmış çıktıyı ekrana göndermek için bir kütüphane fonksiyonudur. Printf () fonksiyonu "stdio.h" başlık dosyasında bildirilmiştir.

Burada, stdio.h bir başlık dosyasıdır (standart giriş çıkış başlık dosyası) ve #include, gerektiğinde kodu başlık dosyasından yapıştırmak için bir ön işlemci yönergesidir. Derleyici printf () fonksiyonuyla karşılaştığında ve stdio.h başlık dosyasını bulamadığında, derleyici hatası gösterir.

return 0; programın "Çıkış durumu" dur. (Exit status)

• Örnek 2 : C Tamsayı Çıkışı (C Integer Output)

```
#include <stdio.h>
int main()
{
    int testInteger = 5;
    printf("Sayi= %d", testInteger);
    return 0;
}
```

Çıktısı :

Sayi= 5

Printf () işlevinde tırnak içinde, "% d" (tam sayı için) biçiminde bir dize var. Biçim dizgisi argümanla eşleşiyorsa (bu durumda testInteger), ekranda görüntülenir.

• Örnek 3 : C Tamsayı Giriş / Çıkış (C Integer Input/Output)

```
#include <stdio.h>
int main()
{
    int testInteger;
    printf("bir tamsayı giriniz: ");
    scanf("%d",&testInteger);
    printf("Sayı = %d",testInteger);
    return 0;
}
```

Çıktısı :

```
bir tamsayı giriniz: 4
Sayı = 4
```

Scanf () işlevi klavyeden biçimlendirilmiş girişi okur. Kullanıcı bir tamsayı girdiğinde, testInteger değişkeninde saklanır.

TestInteger öncesi ‘&’ işaretini not edin; & testInteger, testInteger adresini alır ve değer bu adrese kaydedilir.

• Örnek 4 : C Ondalıklı Sayı Giriş / Çıkış (C Floats Input/Output)

```
#include <stdio.h>
int main()
{
    float f;
```

```
printf("bir sayı giriniz: ");

scanf("%f",&f);
printf("girdiginiz deger= %f", f);
return 0;
}
```

Çıktısı :

```
bir sayı giriniz: 23.45
girdiginiz deger = 23.450000
```

“%f” format dizgisi, ondalıklı sayı durumunda formatlanmış okuma ve görüntüleme için kullanılır.

▪ Örnek 5 : C Karakteri Giriş / Çıkış (C Character I/O)

```
#include <stdio.h>
int main()
{
    char chr;
    printf("bir karakter girin: ");
    scanf("%c",&chr);
    printf("girdiginiz %c.",chr);
    return 0;
}
```

Çıktısı :

```
bir karakter girin: g
girdiginiz g.
```

Karakter tipleri durumunda “%c” format dizgisi kullanılır.

Yukarıdaki programa bir karakter girildiğinde, karakterin kendisi saklanmaz. Bunun yerine, sayısal bir değer ASCII değeri saklanır.

Bu değeri “% c” metin biçimini kullanarak görüntülediğimizde, girilen karakter görüntülenir.

▪ Örnek 6 : C ASCII Kodu (C ASCII Code)

```
#include <stdio.h>
int main()
{
    char chr;
    printf("bir karakter giriniz: ");
    scanf("%c",&chr);

    // % C metin formatı kullanıldığında, karakter türlerinde
    karakter görüntülenir
    printf("girdiginiz karakter %c.\n",chr);

    // % D metin formatı kullanıldığında, karakter türlerinde
    tam sayı görüntülenir.
    printf("ASCII degeri : %c sayisal degeri ise %d.", chr,
chr);
    return 0;
}
```

Çıktısı :

```
bir karakter giriniz: g
girdiginiz karakter g.
```

ASCII degeri : g sayisal degeri ise 103.

Bir karakterin ASCII kodunu biliyorsanız bir karakteri görüntüleyebilirsiniz. Bu, aşağıdaki örnekle gösterilmiştir.

• Örnek 7 : C ASCII Kodu (C ASCII Code)

```
#include <stdio.h>
int main()
{
    int chr = 69;
    printf("69 karakterinin ASCII değeri : %c.",chr);
    return 0;
}
```

Çıktısı :

69 karakterinin ASCII değeri : E.

• Örnek 8 : Ondalıklı ve Tam Sayıların Giriş / Çıkış işlemleri (I/O of Floats and Integers)

```
#include <stdio.h>
int main()
{

    int integer = 9876;
    float decimal = 987.6543;

    printf("4 basamaklı tamsayıyı noktadan sonra 6 basamak yaz: %6d\n", integer);
}
```

```
printf("4 basamaklı tamsayıyı noktadan sonra 3 basamak  
yaz: %3d\n", integer);  
  
printf("Ondalıkli sayıyı noktadan sonra 2 basamak yaz:  
%.2f\n", decimal);  
  
printf("Ondalıkli kısmı 0'a yuvarla: %.f\n", 987.6543);  
  
printf("Ondalıkli sayıyı Üslü biçimde yaz:  
%e\n", 987.6543);  
return 0;  
}
```

Çıktısı :

```
4 basamaklı tamsayıyı noktadan sonra 6 basamak yaz: 9876  
4 basamaklı tamsayıyı noktadan sonra 3 basamak yaz: 9876  
Ondalıkli sayıyı noktadan sonra 2 basamak yaz: 987.65  
Ondalıkli kısmı 0'a yuvarla: 988  
Ondalıkli sayıyı Üslü biçimde yaz: 9.876543e+02
```