

3.Ders : C İşaretçiler ve Fonksiyonlar

C Referansa Göre Ara: İşaretçileri kullanma

Bu eğitici yazıda, örnekleri fonksiyonlara **argüman** olarak **iletmeyi** öğreneceksiniz. Bu teknik, **referans olarak yapılan çağrı** olarak bilinir.

C programlamada, **adresleri fonksiyonlara argüman olarak iletmek de mümkündür.**

Bu adresleri işlem tanımında kabul etmek için işaretçileri kullanabiliriz. Bunun nedeni, **işaretçilerin adresleri depolamak için kullanılmasıdır.** Bir örnek alalım:

```
#include <stdio.h>
void swap(int *n1, int *n2);
int main()
{
    int num1 = 5, num2 = 10;
    // address of num1 and num2 is passed
    swap( &num1, &num2);
    printf("num1 = %d\n", num1);
    printf("num2 = %d", num2);
    return 0;
}
void swap(int* n1, int* n2)
{
    int temp;
    temp = *n1;
```

```
    *n1 = *n2;  
    *n2 = temp;  
}
```

Çıktısı

```
num1 = 10  
num2 = 5
```

Num1 ve num2 adresi, swap (& num1, & num2); kullanılarak swap () işlevine iletilir.

İşaretçiler n1 ve n2, işlev tanımındaki bu argümanları kabul eder.

```
void swap(int* n1, int* n2) {  
    ... ..  
}
```

Swap () işlevinde * n1 ve * n2 değiştirildiğinde, main () işlevinde num1 ve num2 de değişir.

Swap () işlevinin içinde, * n1 ve * n2 değiştirildi. Böylece, num1 ve num2 de değiştirilir.

Bu teknik, C programlamasında referans olarak adlandırılan çağrı olarak bilinir.

Örnek 2: Fonksiyonlardan İşaretçilere Geçmek

```
#include <stdio.h>
void addOne(int* ptr) {
    (*ptr)++; // adding 1 to *ptr
}
int main()
{
    int* p, i = 10;
    p = &i;
    addOne(p);
    printf("%d", *p); // 11
    return 0;
}
```

Burada, p, * p'de depolanan değer başlangıçta 10'dur.

Daha sonra p göstergesini addOne () işlevine geçirdik. Ptr işaretçisi bu adresi addOne () işlevinde alır.

Fonksiyonun içinde, (* ptr) ++; kullanarak ptr'de depolanan değeri 1 arttırdık. Ptr ve p işaretçilerinin her ikisi de aynı adrese sahip olduğundan, main () içindeki * p de 11 olacaktır.