

# 2.Ders : C Yapılar ve İşaretçiler

Bu dersde, C programlamadaki **yapı üyelerine erişmek için işaretçileri kullanmayı** öğreneceksiniz. Ayrıca, yapı türlerinin belleğini **dinamik olarak ayırmayı** da öğreneceksiniz.

İşaretçilerin yapılarla nasıl kullanılabileceğini öğrenmeden önce, bu dersleri kontrol ettiğinizden emin olun:

[C İşaretçiler](#)

[C Yapılar](#)

Bu şekilde yapılar için işaretçiler oluşturabilirsiniz.

```
struct name {
    member1;
    member2;
    .
    .
};
int main()
{
    struct name *ptr, Harry;
}
```

Burada **>struct name>** türünde bir **ptr** işaretçisi yaratılır. Yani, **ptr** yapılandırmak için bir **işaretçidir**.

**Örnek:** İşaretçileri (Pointer) kullanarak üyelere erişim.  
İşaretçileri kullanarak bir yapının üyelerine erişmek için -> operatörünü kullanırız.

```
#include <stdio.h>
struct person
{
    int age;
    float weight;
};
int main()
{
    struct person *personPtr, person1;
    personPtr = &person1;
    printf("Enter age: ");
    scanf("%d", &personPtr->age);
    printf("Enter weight: ");
    scanf("%f", &personPtr->weight);
    printf("Displaying:\n");
    printf("Age: %d\n", personPtr->age);
    printf("weight: %f", personPtr->weight);
    return 0;
}
```

Artık **personPtr** işaretçisini kullanarak **person1** üyelerine erişebilirsiniz.

### **Yapıların dinamik hafıza tahsisi**

Bu bölüme devam etmeden önce, C dinamik bellek tahsisini kontrol etmenizi öneririz.

Bazen, bildirdiğiniz yapı değişkenlerinin sayısı yetersiz olabilir. Çalışma sırasında hafıza ayırmanız gerekebilir. İşte bunu C programlamada nasıl başarabileceğinizi görebilirsiniz.

## Örnek :

```
#include <stdio.h>
#include <stdlib.h>
struct person {
    int age;
    float weight;
    char name[30];
};
int main()
{
    struct person *ptr;
    int i, n;
    printf("Enter the number of persons: ");
    scanf("%d", &n);
    // allocating memory for n numbers of struct person
    ptr = (struct person*) malloc(n * sizeof(struct person));
    for(i = 0; i < n; ++i)
    {
        printf("Enter first name and age respectively: ");
        // To access members of 1st struct person,
        // ptr->name and ptr->age is used
        // To access members of 2nd struct person,
        // (ptr+1)->name and (ptr+1)->age is used
        scanf("%s %d", (ptr+i)->name, &(ptr+i)->age);
    }
    printf("Displaying Information:\n");
    for(i = 0; i < n; ++i)
        printf("Name: %s\tAge: %d\n", (ptr+i)->name,
(ptr+i)->age);
    return 0;
}
```

## Çıktısı :

```
Enter the number of persons: 2
Enter first name and age respectively: Harry 24
Enter first name and age respectively: Gary 32
```

Displaying Information:

Name: Harry      Age: 24

Name: Gary      Age: 32

Yukarıdaki örnekte, n'nin kullanıcı tarafından girildiği yerde n sayısı yapısal değişkenler oluşturulmuştur.

N sayıdaki **person** yapısını hafıza tahsis etmek için :

```
ptr = (struct person*) malloc(n * sizeof(struct person));
```

Daha sonra, **person** elementlerine erişmek için **ptr** işaretçisini kullandık.