

1.Ders : C Yapılar

Bu derste, C Programlama'daki **yapı tipleri** hakkında bilgi edineceksiniz. Örneklerin yardımıyla yapıları tanımlamayı ve kullanmayı öğreneceksiniz. C programlamada, bir yapı , tek bir ad altında bir değişkenler topluluğudur (farklı türlerde olabilir).

C'de yapılar nasıl tanımlanır?

Yapı değişkenleri oluşturabilmeniz için önce **veri tipini** tanımlamanız gerekir. Bir yapı tanımlamak için **struct** anahtar sözcüğü kullanılır.

```
struct structureName
{
    dataType member1;
    dataType member2;
    ...
};
```

İşte bir örnek:

```
struct Person
{
    char name[50];
    int citNo;
    float salary;
};
```

Burada türetilmiş tipte bir yapı elemanı tanımlanmıştır. Şimdi, bu tür değişkenler oluşturabilirsiniz.

Struct değişkenleri oluşturma

Bir yapı tipi bildirildiğinde, depolama veya hafıza tahsis edilmez. Belirli bir yapı tipine ait hafızayı tahsis etmek ve onunla çalışmak için değişkenler yaratmamız gerekiyor.

İşte yapı değişkenlerini şu şekilde yaratıyoruz:

```
struct Person
{
    char name[50];
    int citNo;
    float salary;
};

int main()
{
    struct Person person1, person2, p[20];
    return 0;
}
```

Bir yapı değişkeni oluşturma'nın başka bir yolu:

```
struct Person
{
    char name[50];
    int citNo;
    float salary;
} person1, person2, p[20];
```

Bir yapının erişim üyeleri

Bir yapının üyelerine erişmek için kullanılan iki tür operatör vardır.

. – Üye operatörü

-> – Yapı işaretçisi operatörü

Diyelim ki, 2. kişinin maaşına erişmek istiyorsunuz. İşte bunu nasıl yapabilirsiniz.

person2.salary

Örnek :

```
#include <stdio.h>
struct Distance
{
    int feet;
    float inch;
} dist1, dist2, sum;
int main()
{
    printf("1st distance\n");
    printf("Enter feet: ");
    scanf("%d", &dist1.feet);
    printf("Enter inch: ");
    scanf("%f", &dist1.inch);
    printf("2nd distance\n");
    printf("Enter feet: ");
    scanf("%d", &dist2.feet);
    printf("Enter inch: ");
    scanf("%f", &dist2.inch);
    // adding feet
    sum.feet = dist1.feet + dist2.feet;
    // adding inches
    sum.inch = dist1.inch + dist2.inch;
    // changing feet if inch is greater than 12
    while (sum.inch >= 12)
    {
        ++sum.feet;
        sum.inch = sum.inch - 12;
    }
}
```

```
        printf("Sum of distances = %d\'-%.1f\\\"", sum.feet,
sum.inch);
        return 0;
}
```

Çıktısı :

```
1st distance
Enter feet: 12
Enter inch: 7.9
2nd distance
Enter feet: 2
Enter inch: 9.8
Sum of distances = 15'-5.7"
```

Typedef anahtar sözcüğü

Typedef anahtar sözcüğünü yapılarla kullanabilirsiniz.

```
struct Distance{
    int feet;
    float inch;
};

int main() {
    structure Distance d1, d2;
}
```

eşittir

```
typedef struct Distance{
    int feet;
    float inch;
```

```
} distances;

int main() {
    distances dist1, dist2, sum;
}
```

İç İçe Yapılar

C programlamada bir yapı içinde yapılar oluşturabilirsiniz. Örneğin,

```
struct complex
{
    int imag;
    float real;
};

struct number
{
    struct complex comp;
    int integers;
} num1, num2;
```

Diyelim ki **num2** değişkeninin **imag** değerini 11 olarak ayarlamak istedik. Bunu nasıl yapabiliriz:

```
num2.comp.imag = 11;
```

Neden C'de yapılar?

Bir kişi hakkında bilgi depolamak istediğinizi varsayalım: adı, vatandaşlık numarası ve maaş. Bu bilgiyi saklamak için farklı değişkenler ad, TcNo ve maaş oluşturabilirsiniz.

Birden fazla kişinin bilgilerini saklamanız gerekirse ne olur? Şimdi, kişi başına her bilgi için farklı değişkenler oluşturmanız gerekir: ad1, TcNo1, maaş1, ad2, TcNo2, maaş2, vb.

İlgili tüm bilgilerin tek bir **Kişi** yapısı altında toplanması ve her kişi için kullanılması **daha iyi bir yaklaşım** olacaktır.