

3.Ders : C Yapılar ve Fonksiyonlar

Bu derste, **struct** değişkenlerini bir **fonksiyona argüman olarak iletmeyi** öğreneceksiniz. Örneklerin yardımıyla **yapıyı bir işlevden döndürmeyi** öğreneceksiniz.

Yerleşik türlerin değişkenlerine benzer şekilde, yapı değişkenlerini bir işleve de iletebilirsiniz.

Yapıları fonksiyonlara iletme :

Yapıları fonksiyonlara nasıl ileteceğimizi öğrenmeden önce bu dersleri öğrenmenizi tavsiye ediyoruz.

[C Yapılar](#)

[C Fonksiyonlar](#)

[Kullanıcı tanımlı fonksiyonlar](#)

Burada yapıları bir fonksiyona aktaracağız :

```
#include <stdio.h>
struct student
{
    char name[50];
    int age;
};
// function prototype
void display(struct student s);
int main()
{
    struct student s1;
```

```

    printf("Enter name: ");
    scanf("%[^\\n]*c", s1.name);
    printf("Enter age: ");
    scanf("%d", &s1.age);
    display(s1);    // passing struct as an argument
    return 0;
}
void display(struct student s)
{
    printf("\\nDisplaying information\\n");
    printf("Name: %s", s.name);
    printf("\\nAge: %d", s.age);
}

```

Çıktısı :

Enter name: Bond

Enter age: 13

Displaying information

Name: Bond

Age: 13

Burada, **struct student** türünde bir **struct** değişkeni olan **s1** oluşturulur. Değişken, **display(s1)** kullanılarak **display()** fonksiyonuna iletilir.

Peki bir fonksiyondan yapıyı nasıl geri çağırabiliriz :

```

#include <stdio.h>
struct student
{
    char name[50];
    int age;
}

```

```

};
// function prototype
struct student getInformation();
int main()
{
    struct student s;
    s = getInformation();
    printf("\nDisplaying information\n");
    printf("Name: %s", s.name);
    printf("\nRoll: %d", s.age);
    return 0;
}
struct student getInformation()
{
    struct student s1;
    printf("Enter name: ");
    scanf ("%[^\\n]*c", s1.name);
    printf("Enter age: ");
    scanf ("%d", &s1.age);
    return s1;
}

```

Referansa göre yapıyı iletme :

Yapıları **referansa** göre de iletebilirsiniz. Referansa göre geçiş sırasında, struct değişkenlerinin hafıza adresleri **fonksiyona** iletilir.

```

#include <stdio.h>
typedef struct Complex
{
    float real;
    float imag;
} complex;
void addNumbers(complex c1, complex c2, complex *result);
int main()
{

```

```

    complex c1, c2, result;
    printf("For first number,\n");
    printf("Enter real part: ");
    scanf("%f", &c1.real);
    printf("Enter imaginary part: ");
    scanf("%f", &c1.imag);
    printf("For second number, \n");
    printf("Enter real part: ");
    scanf("%f", &c2.real);
    printf("Enter imaginary part: ");
    scanf("%f", &c2.imag);
    addNumbers(c1, c2, &result);
    printf("\nresult.real = %.1f\n", result.real);
    printf("result.imag = %.1f", result.imag);
    return 0;
}

void addNumbers(complex c1, complex c2, complex *result)
{
    result->real = c1.real + c2.real;
    result->imag = c1.imag + c2.imag;
}

```

Çıktısı :

```

For first number,
Enter real part:  1.1
Enter imaginary part:  -2.4
For second number,
Enter real part:  3.4
Enter imaginary part:  -3.2

```

```

result.real = 4.5
result.imag = -5.6

```

Yukarıdaki programda üç yapı değişkeni c1, c2 ve sonucun adresi **addNumbers ()** fonksiyonuna geçirilir. Burada **sonuç referans ile iletilir.**

AddNumbers () içindeki sonuç değişkeni değiştirildiğinde, **main ()** işlevinin içindeki sonuç değişkeni de buna göre değiştirilir.