



## Konu 1 : “Typedef Struct” Yapısının Kullanımı

### Örnek Kod 1

```
1 // Bu kod "typedef struct" yapısı işlenmiştir.
2 #include<stdio.h> // Kütüphaneler tanımlanır.
3 #include<stdlib.h>
4 #include<string.h>
5
6 int main(){
7     typedef struct{ // "student_info" tipi, "typedef struct" yapısı yardımıyla yerel (local) olarak tanımlanır.
8         char name[20]; // "char" tipinde " name" dizisi tanımlanır.
9         double number; // "double" tipinde " number" değişkeni tanımlanır.
10    }student_info;
11
12    student_info student1,student2 = {"Veli",23}; // Değişkenlerin tipi "student_info" olarak ayarlanır.
13 // "student2" değişkeni önceden tanımlanır.
14
15 // Tanımlanan struct değişkenleri nokta(.) yardımıyla iç değişkenler kullanışır.
16 strcpy(student1.name,"Ali"); // "Student1" değişkeninde bulunan "name" dizisine değer atanır.
17
18 student1.number = 20; // "Student1" değişkeninde bulunan "number" değişkenine değer atanır.
19
20 printf(" %s %.2f\n",student1.name,student1.number);
21 printf(" %s %.2f\n\n",student2.name,student2.number);
22
23 student1 = student2;
24
25 printf(" %s %.2f\n",student1.name,student1.number);
26 printf(" %s %.2f\n",student2.name,student2.number);
27
28 system("pause");
29 return 0;
30 }
```



### Örnek Kod 2

```
1 // Bu kod "typedef struct" yapısı işlenmiştir.
2 #include<stdio.h> // Kütüphaneler tanımlanır.
3 #include<stdlib.h>
4 #include<string.h>
5
6 int main(){
7     int a = 5;
8
9     typedef struct{ // "student_info" tipi, "typedef struct" yapısı yardımıyla yerel (local) olarak tanımlanır.
10        char name[20]; // "char" tipinde " name" dizisi tanımlanır.
11        double number; // "double" tipinde " number" değişkeni tanımlanır.
12    }student_info;
13
14    typedef struct{ // "class_info" tipi, "typedef struct" yapısı yardımıyla yerel (local) olarak tanımlanır.
15        student_info students[3]; // "student_info" tipinde "student" dizisi tanımlanır.
16    }class_info;
17
18    class_info class_1; // "class_info" tipinde "class_1" değişkeni tanımlanır.
19
20    strcpy(class_1.students[0].name,"Ali"); // "class_1" değişkeninde bulunan "students[0]" dizisinin "name" değişkenine değer atanır.
21
22    class_1.students[0].number = 5; // "class_1" değişkeninde bulunan "students[0]" dizisinin "number" değişkenine değer atanır.
23
24
25    printf(" %s %.2f\n",class_1.students[0].name, class_1.students[0].number);
26
27    system("pause");
28    return 0;
29 }
30 }
```



## Konu 2 : Dosya İşleme

### Örnek Kod 1

```
1 // Bu kod "Dosya İşleme " konusu işlenmiştir.
2 #include<stdio.h> // Kütüphaneler tanımlanır.
3 #include<stdlib.h>
4
5 int main(){
6     FILE *dosya; // FILE tipinde bir gösterici (pointer) tanımlanır.
7
8     dosya = fopen("ogrenci.txt","r"); // "fopen(file open)" fonksiyonu ile açılan txt dosyası, göstericiye eşitlenir.
9     // "r(read)" ile dosyadan veri okunacağı tanımlanır.
10
11     char isim[100],okul[50],bolum[50];
12
13     printf("Ad-Soyad :"); gets(isim); // "gets" fonksiyonu ile dizi ekrana bastırılır.( printf ile aynıdır)
14     printf("Okulunuz :"); gets(okul);
15     printf("Bolumunuz :"); gets(bolum);
16
17     fprintf(dosya , "%s\n%s\nf%s\n\n", isim,okul,bolum); // " fprintf (file printf)" fonksiyonu kullanılırken;
18     // önce göstericinin ismi yazılır.
19     // ikinci kısımda yazılmak istenen dizi yazılır.
20     // (Varsa) üçüncü kısımda değişkenlerin isimleri yazılır.
21
22     fclose(dosya); // Açılan txt dosyası " fclose (file close)" fonksiyonu ile kapatılır.
23
24     system("pause");
25     return 0;
26 }
```



### Örnek Kod 2

```
1 // Bu kod "Dosya İşleme " konusu işlenmiştir.
2 #include<stdio.h> // Kütüphaneler tanımlanır.
3 #include<stdlib.h>
4
5 int main(){
6     FILE *dosya;
7     char k1[10];
8
9     int sayi;
10
11     dosya =fopen("deneme.txt","r"); // "fopen(file open)" fonksiyonu ile açılan txt dosyası, göstericiye eşitlenir.
12     // "r(read)" ile dosyadan veri okunacağı tanımlanır.
13
14     fscanf(dosya,"%s",&k1); // " fscanf (file scanf)" fonksiyonu kullanılırken;
15     // önce göstericinin ismi yazılır.
16     // ikinci kısımda okunmak istenen veri tipi yazılır.
17     // üçüncü kısımda adres biti ile değişkenlerin isimleri yazılır.( "scanf" fonksiyonundaki gibi)
18
19     fscanf(dosya,"%d",&sayi);
20
21     printf("%s %d",k1,sayi);
22
23     fclose(dosya); // Açılan txt dosyası " fclose (file close)" fonksiyonu ile kapatılır.
24
25     system("pause");
26     return 0;
27 }
```



ROBOT TOPLULUĞU

### Konu 3 : Göstericiler( Pointer )

#### Örnek Kod 1

```
1 // Bu kod "Göstericiler ( Pointer )" konusu işlenmiştir.
2 #include<stdio.h> // Kütüphaneler tanımlanır.
3 #include<stdlib.h>
4 // Göstericilerde değer yerine değişkenlerin adresi tutulur.
5 int main(){
6     int sayi=5; // "sayi" isminde değişken tanımlanır ve 5'e eşitlenir.
7     int *p1; // "p1" isminde gösterici tanımlanır.
8
9     p1=&sayi; // "sayi" değişkeninin adresi "p1" göstericisine eşitlenir.
10
11     int sayi2=*p1; // "p1" göstericisinin gösterdiği(point ettiği)
12 // değişkenin değeri "sayi2" değişkenine eşitlenir.
13
14     printf("sayi : %d\n",sayi); // "sayi" değişkeninin değeri ekrana bastırılır.
15     printf("sayi degiskeninin adresi(decimal) : %d\n",&sayi); // "sayi" değişkeninin adresi (decimal) ekrana bastırılır.
16     printf("sayi degiskeninin adresi(hexadecimal) : %p\n",&sayi); // "sayi" değişkeninin adresi(hexadecimal) ekrana bastırılır.
17     printf("p1 gostericisinin degeri : %d\n",p1); // "p1" göstericisinin değeri (decimal) ekrana bastırılır.
18 // Yukardaki örnekten "p1" göstericisinin içinde
19 // "sayi" değişkeninin adresinin tutulduğu gözlemlenebilir.
20     printf("p1 degiskeninin adresi(decimal) : %d\n",&p1); // "p1" göstericisinin adresi (decimal) ekrana bastırılır.
21     printf("sayi2 : %d\n",sayi2);
22
23     system("pause");
24     return 0;
25 }
```



#### Örnek Kod 2

```
1 // Bu kod "Göstericiler ( Pointer ) ve Diziler" konusu işlenmiştir.
2 #include<stdio.h> // Kütüphaneler tanımlanır.
3 #include<stdlib.h>
4
5 int main(){ // Her dizi bir göstericidir.
6     int dizi[]={5,4,8,9,11}; // Göstericilerin içinde değer yerine adres kaydedilir.
7     int *ptr; // "ptr" isminde gösterici tanımlanır.
8
9     ptr=dizi; // "dizi" değişkeni "ptr" değişkenine eşitlenir.
10
11     printf("Dizinin 2. elemanı :%d -- adresi : %p\n\n",dizi[1],(ptr+1)); // " %p " kullanımı ile sayılar heksadesimal olarak ekrana bastırılır.
12     printf("Dizinin 0. elemanı :%d\n\n",*ptr); // Aynı : *ptr == ptr[0]
13     printf("Dizinin 4. elemanı :%d\n\n",*(ptr+3)); // Aynı : *(ptr+3) == ptr[3]
14     printf("Dizinin 5. elemanı :%d\n\n",*(dizi+4)); // Aynı : *(dizi+4) == ptr[4]
15
16     for(int i=0;i<5;i++){
17         printf("dizi[%d]:%d - adresi %p \n",i,*(ptr+i),(ptr+i));
18     }
19
20     system("pause");
21     return 0;
22 }
```



ROBOT TOPLULUĞU