

5.Ders : C Depolama Sınıfı (C Storage Class)

Bu eğitimde **Lokal** ve **Global** değişkenlerin **kapsamı** ve **ömrü** hakkında bilgi edineceksiniz. Ayrıca, **statik** ve **register** değişkenlerini öğreneceksiniz.

Bir değişkenin kapsamı ve ömrü

İçindekiler

- # Değişken özellikleri (Variable properties)
- # Lokal değişken (Local variable)
- # Global değişken (Global variable)
 - *Örnek 1: Global değişken*
- # Kayıt değişkeni (Register variable) (**artık kullanılmamaktadır**)
- # Statik değişken (Static variable)
 - *Örnek 2: Statik değişken*

C programlamanın her değişkeni iki özelliğe sahiptir: type ve storage class.

type, bir değişkenin **veri türünü** ifade eder. **Storage** (depolama) sınıfı, bir değişkenin **kapsamını**, **görünürlüğünü** ve **ömrünü** belirler.

4 tip depolama sınıfı vardır:

automatic
external
static
register

Lokal değişken (Local variable)

Fonksiyon içinde bildirilen değişkenler **otomatik** veya **lokal** değişkenlerdir.

Lokal değişkenler yalnızca bildirildiği Fonksiyonun içinde bulunur. Fonksiyondan çıkıldığında, lokal değişkenler yok edilir.

```
int main() {  
    int n; // n, main () fonksiyonunda lokal bir değişkendir  
    ... ..  
}  
  
void func() {  
    int n1; // n1 func () fonksiyonunda lokaldir  
}
```

Yukarıdaki kodda, func() bitişi ile n1 imha edilir. Aynı şekilde, main () bitince n yok olur.

Global değişken (Global variable)

Tüm Fonksiyonların dışında bildirilen değişkenler, **harici** veya

global değişkenler olarak bilinir. Programın içindeki herhangi bir fonksiyondan erişilebilirler.

Örnek 1: Global Değişken

```
#include <stdio.h>
void display();

int n = 5; // Global Değişken

int main()
{
    ++n;      // n değişkeni main () fonksiyonunda bildirilmez.
    display();
    return 0;
}

void display()
{
    ++n;      // display () fonksiyonunda n değişkeni
    bildirilmez.
    printf("n = %d", n);
}
```

Çıktısı :

n = 7

Diyelim ki global bir değişken dosya1'da bildirildi. Bu değişkeni farklı bir dosya2 dosyasında kullanmaya çalışırsanız, derleyici hata verecektir. Bu sorunu çözmek için, **extern** anahtar sözcüğü, *dış değişkenin başka bir dosyada bildirildiğini belirtmek* için dosya2'de kullanılır.

Kayıt değişkeni (Register variable)

Register anahtar sözcüğü register değişkenlerini bildirmek için kullanılır. Kayıt değişkenlerinin yerel değişkenlerden daha hızlı olması gerekiyordu.

Ancak, modern derleyiciler kod optimizasyonunda çok iyidir ve kayıt değişkenlerinin kullanılmasının programınızı daha hızlı hale getirme olasılığı çok düşüktür.

Verilen uygulama için kodu nasıl optimize edeceğinizi bildiğiniz gömülü sistemler üzerinde çalışmadığınız sürece, kayıt değişkenlerinin kullanımı yoktur.

Statik değişken (Static variable)

Statik bir anahtar sözcük kullanılarak statik bir değişken bildirilir. Örneğin;

```
static int i;
```

Statik değişkenin değeri programın sonuna kadar devam eder.

Örnek 2: Statik Değişken

```
#include <stdio.h>
void display();

int main()
{
    display();
    display();
}
void display()
{
    static int c = 0;
    printf("%d  ",c);
    c += 5;
}
```

Çıktısı :

0 5

İlk fonksiyon çağrısı sırasında, c'nin değeri 0'a eşittir. Sonra değeri 5'e çıkarılır.

İkinci fonksiyon çağrısı sırasında c değişkeni tekrar 0 olarak başlatılmaz. Çünkü c statik bir değişkendir. Böylece ekranda 5 görüntülenir.