

3.Ders : C Değişkenler ve Sabitler

C Programlama Sabitleri ve Değişkenleri.

Bu derste, C programlamada değişkenler, değişken isimlendirme kuralları, sabitler ve farklı sabit tipleri hakkında bilgi edineceksiniz.

İçindekiler

* Değişkenler (Variables)

– Bir değişkeni adlandırmak için kurallar

* Sabitler (Constants)

– Tamsayılar (Integers)

– Ondalıklı ve noktalı sayılar (Floating-point Numbers)

– Karakter Sabiti (Character Constants)

– Kaçış dizileri (Escape Sequences)

– Dize değişmezleri (String Literals)

– Numaralandırma (Enumerations)

Değişkenler (Variables)

Programlamada bir değişken, veriyi tutmak için bir konteynerdir (depolama alanı).

Depolama alanını belirtmek için her değişkene benzersiz bir ad

([tanımlayıcı](#)) verilmelidir. Değişken isimleri sadece bir hafıza konumunun sembolik gösterimidir. Örneğin:

```
int playerScore = 95;
```

Burada playerScore, bir tamsayı tipi değişkenidir. Burada, değişkene bir tamsayı değeri 95 atanır.

Bir değişkenin değeri değiştirilebilir, dolayısıyla isim değişkeni değiştirilebilir.

```
char ch = 'a';  
ch = 'l';
```

Bir değişkeni adlandırmak için kurallar

- 1 – Bir değişken isminde harfler (hem büyük hem küçük harf), yalnızca rakam ve alt çizgi olabilir.
- 2 – Bir değişkenin ilk harfi bir harf veya alt çizgi olmalıdır.
- 3 – Bir değişken adının (tanımlayıcı) ne kadar süreceği konusunda hiçbir kural yoktur. Ancak değişken ismi 31 karakterden uzunsa bazı derleyicilerde problem yaşayabilirsiniz.

Not: Değişkenlere her zaman anlamlı adlar vermeye çalışmalısınız. Örneğin: firstName, fn'den daha iyi bir değişken adıdır.

C kurallı yazılmış bir dildir. Bu, bildirildikten sonra değişken türünün değiştirilemeyeceği anlamına gelir. Örneğin:

```
int number = 5;           // tamsayı değişkeni
number = 5.5;             // hatalı
double number;            // hatalı
```

Burada sayı değişkeninin türü int'dir. Bu değişkene ondalık değeri 5.5 atayamazsınız. Ayrıca, değişkenin türünü double olarak yeniden tanımlayamazsınız. Bu arada, C'deki ondalık değerleri saklamak için türünü double ya da float olarak bildirmeniz gerekir.

Sabitler (Constants)

Sabit değeri, bir programda değiştirilemeyen bir değerdir .
Örneğin: 1, 2.5, 'c' vb.

Burada, 1, 2.5 ve 'c' değişmez sabitleridir. Bu terimlere farklı değerler atayamazsınız.

C programlamasında değiştirilemez değişkenler de oluşturabilirsiniz. Örneğin:

```
const double PI = 3.14;
```

Uyarı !!! anahtar kelimeye const ekledik.

Burada PI sembolik bir sabittir. Bu aslında bir değişken, ancak değeri değiştirilemez.

```
const double PI = 3.14;  
PI = 2.9;           //Hatalı
```

Aşağıda, C'de kullanabileceğiniz farklı sabit türleri bulunmaktadır:

Tamsayılar (Integers)

Bir tamsayı, kesirli veya üstel bir kısmı olmayan sayısal bir sabittir . C programlamada üç tip tamsayı sabiti vardır:

- onluk sabiti (taban 10)
- sekizli sabiti (taban 8)
- onaltılık sabit (taban 16)

Örneğin:

Decimal constants: 0, -9, 22 etc

Octal constants: 021, 077, 033 etc

Hexadecimal constants: 0x7f, 0x2a, 0x521 etc

C programlamada, sekizli 0 ile başlar ve onaltılık 0x ile başlar.

Ondalıklı ve noktalı sayılar (Floating-point Numbers)

Ondalıklı sayı sabiti, kesirli bir sayı veya bir üs formuna sahip sayısal bir sabittir. Örneğin:

-2.0
0.0000234
-0.22E-5

Not: E-5 = 10^{-5}

Karakter Sabiti (Character Constants)

Tek bir tırnak içine tek bir karakter ekleyerek bir karakter sabiti oluşturulur. Örneğin: 'a', 'm', 'F', '2', '}' vb;

Kaçış dizileri (Escape Sequences)

Bazen, yazılamayan veya C programlamada özel bir anlamı olan karakterleri kullanmamak gerekir. Örneğin: newline (enter), sekme, soru işareti vb. Bu karakterleri kullanmak için kaçış dizisi kullanılır.

Örneğin: \ n newline için kullanılır. Ters eğik çizgi \,

karakterlerin derleyici tarafından işlendiği normal yoldan kaçışa neden olur.

Kaçış Dizileri	Karakter
\b	Backspace
\f	Form feed
\n	Newline
\r	Return
\t	Horizontal tab
\v	Vertical tab
\\	Backslash
\'	Single quotation mark
\"	Double quotation mark
\?	Question mark
\0	Null character

Dize değişmezleri (String Literals)

Bir dize değişmezi, çift tırnak içine alınmış karakter dizisidir. Örneğin:

"good"	//dize sabiti - string constant
""	//boş dize sabiti
" "	//altı tane boşluğun dize sabiti
"x"	//tek karakterli dize sabiti.
"Earth is round\n"	//yazıdan sonra yeni satır ekler

Numaralandırma (Enumerations)

Anahtar sözcük numaralandırması, numaralandırma türlerini tanımlamak için kullanılır. Örneğin:

```
enum renk {sari, yesil, siyah, beyaz};
```

Burada renk değişkendir ve sarı, yeşil, siyah ve beyaz sırasıyla 0, 1, 2 ve 3 değerlerine sahip numaralandırma sabitleridir.

Ayrıca #define kullanarak sembolik sabitleri de tanımlayabilirsiniz.

Daha fazla bilgi edinmek için bu örnekleri inceleyin:

- [C ile Ekrana Kullanıcı Tarafından Girilen Tamsayıyı Basma Programı](#)
- [C ile Kullanıcı Tarafından Girilen İki Tam Sayıyı Toplama Programı](#)
- [C ile Kullanıcı Tarafından Girilen İki Ondalıklı Sayıyı Çarpma Programı](#)