C ile Programlama Doküman 4

(Sabitler/Constants)

Program içerisinde doğrudan yazılan sayılara sabit denir. Yalnızca değişkenlerin değil sabitlerin de türleri vardır. Bir sabitin türü sayının niceliği ve sonuna getirilen eklere bağlıdır. Sabitlere ilişkin kurallar şöyledir:

1. Sayı nokta içermiyorsa, sonuna herhangi bir ek almamışsa ve int türü sınırları içerisindeyse int türdendir.

123 => int

1. Sayı nokta içeriyorsa ve sayının sonunda küçük veya büyük L varsa (L veya l), sabit long türdendir. Sayı long türünün sınırları dışındaysa error oluşur.

100000000L => long

1. Sayı nokta içeriyorsa ve sonuna herhangi bir ek almamışsa double türdendir.

25.6 => double

1. Sayı nokta içeriyorsa ve sonuna büyük veya küçük f(F veya f) almışsa float türündendir.

98.45F => float

1. Sayı nokta içermiyorsa ve sonuna büyük(U) veya küçük (u) almışsa unsigned int(unsigned) türündendir.

10u => unsigned int

1. Sayı nokta içermiyorsa ve sonuna büyük(LU) veya küçük (lu) almışsa unsigned long türündendir.

45lu => unsigned long

1. Sayı nokta içermiyorsa ve sonuna büyük(LLU) veya küçük llu almışsa unsigned long long türündendir.

89llu => unsigned long long

1. Sayı nokta içermiyorsa ve sonuna büyük(LL) veya küçük(ll) almışsa long long türündendir.

6189LL => long long int

1. Sayı nokta içeriyorsa ve sonuna büyük(L) veya küçük(l) almışsa long double türdendir.

234.45L => long double

1. Tek tırnak içerisinde karakterler char türden sabit belirtir. Örneğin:

‘a’ => char

Tek tırnak içerisine yalnızca tek bir karakter yerleştirilebilir. Örneğin:

‘ab’ => error değil ancak yanlış bir kullanım bu durumda her zaman sondaki karakter alınır. Örneğin bu örnek için b veya b’nin ascii tablosundaki değeri alınacaktır. Bu syntax hatası olmasa da böyle bir kullanım kesinlikle tavsiye edilmez.

Tek tırnak içerisindeki karakterler aslında o karakterler aslında karakterin UNICODE tablodaki sıra numarasını belirten sayılardır. Örneğin:

‘a’ + 1 => 98 elde edilir.

**Anahtar Notlar:** C’de int türden bir sabit char türünün sınırları içerisinde kalıyorsa char türden bir değişkene doğrudan atanabilir. Atanan bu sayı UNICODE tablosundaki sıra numarası olarak ele alınır.

Bazı karakterlerin görüntü karşılığı yoktur. Bunlar ekrana yazdırılmak istenirse bir olay gerçekleşir. Bu tür karakterlere bastırılamayan karakterler(nonprintable-nongraphic) karakterler denilmektedir. Bastırılamayan karakterlerin bazıları özel bir yöntemle belirtilmektedir. Bu karakterlere escape sequence karakterler denilmektedir. Tek tırnak içerisinde önce ters bölü sonra bazı özel karakterler bazı UNICODE karakterleri temsil eder. C programlama dilinde bu karakterlerin listesi aşağıdaki gibidir:

‘\b’ => backspace

‘\t’ => horizontal tab

‘\n’ => line feed/new line

‘\f’ => form feed

‘\r’ => carriage return

‘\0’ => null character

Ters bölü karakterleri iki tırnak içerisinde (string atomlarında) tek bir karakter olarak kullanılır. Örneğin :

#include <stdio.h>

int main(void)

{

printf(“C:\\user\\Berke”);//C:\user\Berke

}

Ters bölü karakterinin kendisi ‘\\’ biçiminde belirtilir. Tek tırnak karakterinin kendisi çift tırnak içerisinde doğrudan kullanılabilir. Ancak karakter sabiti olarak ‘\’ biçiminde yazılmalıdır. Örneğin:

#include <stdio.h>

int main(void)

{

char ch = ‘\’;

printf(“’ankara’istanbul”);

printf(“%c\n”,ch);

}

Çift tırnak karakterinin kendisi çift tırnak içerisinde doğrudan kullanılamaz. Çift tırnak karakteri çift tırnak içerisinde ‘\’ biçiminde yazılmalıdır. Çift tırnak karakteri tek tırnak içerisinde doğrudan kullanılabilir. Çift tırnak karakter sabiti ayrıca \” şeklinde de yazılabilir. Örneğin:

#include <stdio.h>

int main(void)

{

char ch1 = ‘”’,ch2 = ‘\”’;

printf(“\”ankara\””);

printf(“%c\n”,ch1);

printf(“%c\n”,ch2);

}

**Tamsayıların Bazı Sayı Sistemlerde Yazılışı**

Tamsayı sabitleri çeşitli sayı tabanlarında (sistemlerinde) yazılabilmektedir:

-Onluk Sistem(Decimal) : Doğrudan rakamlar (0,1,2…..,9) kullanarak yazılan sabitler. Örneğin:

1008 => decimal

- Onaltılık Sistem (Hexadecimal) : Sabit onaltılık tabanda yazılacaksa önce sabitin önüne 0x veya 0X (sıfır ve x/X) yazılmalıdır. Daha sonra 0 ve 9 arasındaki rakamlar ve A, B, C, D, F karakterleri kullanılabilir. Bu karakterler küçük harf olarak da yazılabilir. Örneğin:

0x1FC0 →hexadecimal

0x1fc0 → hexadecimal

- Sekizlik Sistem (Octal): Sabit sekizlik tabanda yazılacaksa önce sabitin önüne 0(sıfır) yazılır daha sonra 0 ve 7 (7 dahil) arasındaki rakamlar kullanılabilir.

Örneğin: 067 → octal

**Karakter Sabitlerini \u ile Yazılması**

Karakter sabitleri tek tırnak içerisinde \u ile UNICODE tablosundaki sıra numarası kullanılarak yazılabilir. Örneğin:

“\u00F1” => \u ile hexadecimal yazılış

Burada 00F1 onaltılık sistemde belirtilmiştir ve ilgili karakterin UNICODE tablosundaki sıra numarasıdır. Örneğin, bu sabit bir değişkene atanacaksa 00F1 sayısı değişkene aktarılır.

**Anahtar Notlar:** C programlama dilinde bazı programlama dillerinin aksine sabitleri yazarken underscore karakteri kullanılamaz. Örneğin:

int x = 1\_000\_000; //error