**Switch Deyimi**

Switch deyiminin genel biçimi şöyledir:

switch(<tamsayı türünden ifade>){

case <sabit ifadesi(s,i)>

<deyim(ler)>

break;

case <s,i>

<deyim(ler)>

break;

//…

default:

//….

break;

}

switch anahtar sözcüğünden sonra parantezler içerisinde bir ifade bulunmak zorundadır. switch deyimi case bölümlerinden oluşur. case anahtar sözcüğünden sonra bir sabit ifadesi bulunmak zorundadır. switch deyimi default bölüme sahip olabilir. Ancak bu bir zorunluluk değildir default kısım if deyiminin else kısmına benzetilebilir.

**Anahtar Notlar :** Yalnızca operatör ve sabitlerden oluşan ifadelere sabit ifadeleri denir. Tek başına sabitler de sabit ifadesidir. Örneğin:  
8

10 – 8 \* 11

12 – 3

Birer sabit ifadesidir. C’de bazı durumlarda sabit ifadesi kullanmak zorunludur. Sabit ifadelerinin net sayısal değerleri derleme aşamasında belirlenip(constant folding) byte koda yazılmaktadır.

Switch deyimi şöyle çalışır:

Önce switch deyimi içerisindeki ifadenin sayısal değeri hesaplanır.Sonra bu değere tam eşit olan case bölümü var mı diye tek tek bakılır. Eğer böyle bir case bölümü varsa akış o bölüme aktarılır. break deyimi switch deyimini sonlandırmak için de kullanılabilmektedir. Eğer switch parantezi içerisindeki ifade ile aynı değerde bir case bölümü yoksa fakat default bölüm varsa akış default bölümüne aktarılır. default bölümü yoksa akış switch deyiminin sonundan devam eder. Örneğin:

#include <stdio.h>

int main(void)

{

printf("Bir sayi giriniz?\n");

int val;

scanf("%d",&val);

switch (val) {

case 1:

printf("Bir\n");

break;

case 2:

printf("iki\n");

break;

default:

printf("Gecersiz sayi\n");

}

}

Switch deyimi içerisinde aynı değere sahip birden fazla case bölümü error oluşturur. Bunun nedeni iki case bölümü de aynı anlama geldiği için bir ambiguity(iki anlamlılık) olmasından kaynaklanmaktadır. Örneğin aşağıdaki kodda bir syntax error söz konusudur:

#include <stdio.h>

int main(void)

{

printf("Bir sayi giriniz?\n");

int val;

scanf("%d",&val);

switch (val) {

case 1:

printf("Bir\n");

break;

case 2:

printf("iki\n");

break;

case 6 / 3:

break;

default:

printf("Gecersiz sayi\n");

}

}

Sabit ifadelerinin değerleri derleme zamanında hesaplanabildiğinden case 2: ile case 6 / 3: bölümleri aynı değerde olacaktır. Case bölümlerinin sıralı olarak dizilmesi ya da default bölümün sonda olması zorunlu değildir. Ancak hem performans hem de okunabilirlik açısından mümkünse gelme olasılığı daha yüksek olan değerler üste, default bölümü tüm case bölümlerinin altına yazılmalıdır. Switch parantezi içerisinde ifade ve case ifadeleri tamsayı türlerine ilişkin olmak zorundadır. Örneğin:

#include <stdio.h>

int main(void)

{

printf("Bir sayi giriniz?\n");

double val;

scanf("%lf",&val);

switch (val) { //error

case 1:

printf("Bir\n");

break;

case 2:

printf("iki\n");

break;

default:

printf("Gecersiz sayi\n");

}

}

Burada val double türden olduğundan geçersizdir. Ancak tabi ki float veya long double türden de olsa gene error olacaktır.

C’de switch deyiminin aşağı düşme (fall through) özelliği bulunmaktadır. Buna göre akış herhangi bir case bölümüne geldiğinde o akışı yönlendirecek herhangi bir deyim görene kadar tüm deyimler çalıştırılacakır. Örneğin:

#include <stdio.h>

int main(void)

{

printf("Bir sayi giriniz?\n");

int val;

scanf("%d",&val);

switch (val) {

case 1:

printf("Bir\n");

break;

case 2:

printf("iki\n");

case 3:

printf("Uc\n");

case 4:

printf("Dort\n");

default:

printf("Gecersiz sayi\n");

}

}

Burada örneğin kullanıcı klavyeden 2 (iki) değerini girerse aşağı düşme özellğinden dolayı case2, case 3 ve case 4 bölümlerinin hepsine ilişkin deyimler çalıştırılacaktır. Aşağı düşmeyi engellemek için break deyimi kullanılabilir. Bununla birlikte akışı aşağı düşürmeyecek herhangi bir deyim de (return, continue, etiktetli break vb.) kullanılabilir. Bazen programcı aşağı düşme özellğinden faydalanabilir. Örneğin:

#include <stdio.h>

int main(void)

{

printf("Secenek giriniz?\n");

char ch;

scanf("%c",&ch);

switch (ch) {

case 'e':

case 'E':

printf("Ekle\n");

break;

case 's':

case 'S':

printf("Sil\n");

break;

default:

printf("Gecersiz Menu\n");

} }

Burada kullanıcı örneğin E veya e girdiğinde aşağı düşme özellği sayesinde aynı işlem yapılacaktır. Bu işlem en kısa bu şekilde yapılabilir.

Örnek Program:

Basit bir menü uygulaması:

#include <stdio.h>

void displayMenu() {

printf("1.Add\n");

printf("2.Update\n");

printf("3.Delete\n");

printf("4.List\n");

printf("5.Exit\n");

printf("Option:");

}

int main() {

for (;;) {

displayMenu();

int option;

scanf("%d", &option);

switch (option) {

case 1:

printf("Add\n");

break;

case 2:

printf("Update\n");

break;

case 3:

printf("Delete\n");

break;

case 4:

printf("List\n");

break;

case 5:

printf("Thank you\n");

return 0;

default:

printf("Invalid option\n");

}

}

}

**Anahtar Notlar:** break deyiminin kullanımı yalnızca switch deyiminde ve döngü deyimleri içerisinde yapılabilir. Programın herhangi başka yerinde kullanımı error oluşturur.