# C switch...case Statement

In this tutorial, you will learn to write a switch statement in C programming (with an example).

The if..else..if ladder allows you to execute a block code among many alternatives. If you are checking on the value of a single variable in if...else...if, it is better to use switch statement.

The switch statement is often faster than nested if...else (not always). Also, the syntax of switch statement is cleaner and easy to understand.

## Syntax of switch...case

switch (n)

​{

case constant1:

// code to be executed if n is equal to constant1;

break;

case constant2:

// code to be executed if n is equal to constant2;

break;

.

.

.

default:

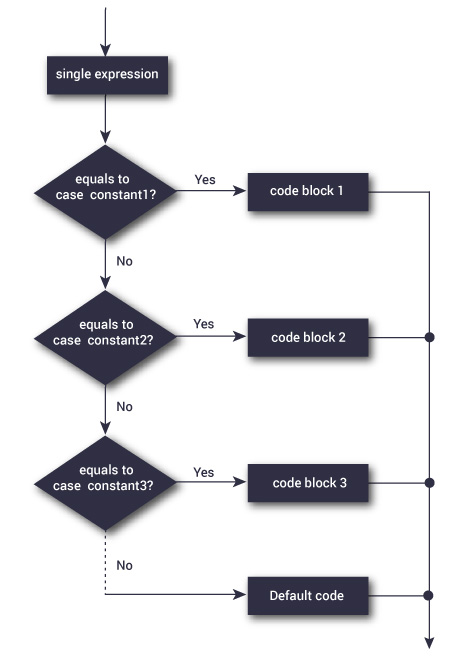
// code to be executed if n doesn't match any constant

}

When a case constant is found that matches the switch expression, control of the program passes to the block of code associated with that case.

Suppose, the value of n is equal to constant2. The compiler executes the statements after case constant2: until break is encountered. When break statement is encountered, switch statement terminates.

### switch Statement Flowchart



### Example: switch Statement

1. // Program to create a simple calculator
2. #include <stdio.h>
3. int main() {
4. char operator;
5. double firstNumber,secondNumber;
6. printf("Enter an operator (+, -, \*, /): ");
7. scanf("%c", &operator);
8. printf("Enter two operands: ");
9. scanf("%lf %lf",&firstNumber, &secondNumber);
10. switch(operator)
11. {
12. case '+':
13. printf("%.1lf + %.1lf = %.1lf",firstNumber, secondNumber, firstNumber+secondNumber);
14. break;
15. case '-':
16. printf("%.1lf - %.1lf = %.1lf",firstNumber, secondNumber, firstNumber-secondNumber);
17. break;
18. case '\*':
19. printf("%.1lf \* %.1lf = %.1lf",firstNumber, secondNumber, firstNumber\*secondNumber);
20. break;
21. case '/':
22. printf("%.1lf / %.1lf = %.1lf",firstNumber, secondNumber, firstNumber/secondNumber);
23. break;
24. // operator doesn't match any case constant (+, -, \*, /)
25. default:
26. printf("Error! operator is not correct");
27. }
28. return 0;
29. }

**Output**

Enter an operator (+, -, \*,): -

Enter two operands: 32.5

12.4

32.5 - 12.4 = 20.1

The - operator entered by the user is stored in operator variable. And, two operands 32.5 and 12.4 are stored in variables firstNumber and secondNumber respectively.

Then, control of the program jumps to

printf("%.1lf / %.1lf = %.1lf",firstNumber, secondNumber, firstNumber/firstNumber);

Finally, the break statement terminates the switch statement.