

Switch Case



Control/ Control flow statements

- In C, the semicolon is a statement terminator.
- Braces { and } are used to group declarations and statements together into a *compound statement*, or *block*, so that they are syntactically equivalent to a single statement.
- The braces that surround the statements of a 'main' function are one obvious example; braces around multiple statements after an if, else, while, or for are another.
- There is no semicolon after the right brace that ends a block.
- C programming language assumes any **non-zero** and **non-null** values as **true**, and if it is either **zero** or **null**, then it is assumed as **false** value.

Control Instruction

- To alter our actions by circumstances.
- If the weather is fine, then I will go for a stroll
- If the highway is busy I would take a diversion
- Notice that all these decisions depend on some condition being met
- C language too must be able to perform different sets of actions depending on the circumstances
- C has three major decision making or conditional statements.
 1. **if-else** statement
 2. **switch** statement
 3. **conditional operator(?:)** statement

The Case Control Structure - switch

- Make a choice between a number of alternatives rather than only one or two.
- Choice is more complicated than merely selecting between two alternatives.
- It is same like if else-if ladder statement.
- Rather than using a series of **if statements you can use switch statement.**
- This statement can be used as multi-way decision statement.
- The switch statement tests the value of a given variable or expression against a list of case values and when a match is found, a block of statements associated with that case is executed.

Switch Syntax

```
■ switch (expression or variable) {  
    case constant 1 :  
        do this ;  
        break;  
    case constant 2 :  
        do this ;  
        break;  
    case constant 3 :  
        do this ;  
        break;  
    default :  
        do this ;  
}
```

Decisions Using switch Syntax

```
switch(expression) {
```

```
    case constant-expression :
```

```
        statement(s);
```

```
        break; /* optional */
```

```
    case constant-expression :
```

```
        statement(s);
```

```
        break; /* optional */
```

```
    /* you can have any number of case statements */
```

```
    default : /* Optional */
```

```
        statement(s);
```

```
}
```

The following rules apply to a switch statement –

- The switch expression must be of **integer or character type**.
- You can have any number of case statements within a switch. Each case is followed by the value to be compared to and a colon.
- All case expressions must be different.
- The constant-expression for a case must be the same data type as the variable or expression in the switch, and it must be a constant or a literal.
- When the variable being switched on is equal to a case, the statements following that case will execute until a break statement is reached.
- When a break statement is reached, the switch terminates, and the flow of control jumps to the next line following the switch statement.
- Not every case needs to contain a break. If no break appears, the flow of control will fall through to subsequent cases until a break is reached.
- A switch statement can have an optional default case. The default case can be used for performing a task when none of the cases is true. No break is needed in the default case.

Rules table of switch

Valid Switch	Invalid Switch	Valid Case	Invalid Case
switch(a)	switch(2.3)	case 3;	case 2.5;
switch(a>b)	switch(a+2.5)	case 'a';	case x;
switch(a+b-2)		case 1+2;	case x+2;
switch(func(a,b))		case 'a'>'b';	case 1,2,3;

What would be the output of the following program:

```
// There is no break in all cases
#include <stdio.h>

int main()
{
    int x = 2;
    switch(x)
    {
        case 1: printf("Choice is 1\n");
        case 2: printf("Choice is 2\n");
        case 3: printf("Choice is 3\n");
        default: printf("Choice other than
1, 2 and 3\n");
    }

    return 0;
}
```

}
Output:

Choice is 2

Choice is 3

Choice other than 1, 2 and 3

What would be the output of the following program:

```
char grade = 'B';
switch(grade)
{
    case 'A' :
        printf("Excellent!\n" );
        break;
    case 'B' :
        printf("Well done\n" );
        break;
    case 'D' :
        printf("You passed\n" );
        break;
    default :
        printf("Invalid grade\n" );
}
printf("Your grade is  %c\n", grade );
```

What would be the output of the following program:

```
#include <stdio.h>
```

```
int main() {
```

```
    int x = 2;
```

```
    switch (x) {
```

```
        case 1: printf("Choice is 1\n");
```

```
        case 2: printf("Choice is 2\n");
```

```
        case 3: printf("Choice is 3\n");
```

```
        case 4: printf("Choice is 4\n");
```

```
            break;
```

```
        default: printf("other\n");
```

```
            break;
```

```
    }
```

```
    printf("After Switch");
```

```
    return 0;
```

```
}
```

Output:

Choice is 2

Choice is 3

Choice is 4

After Switch

What would be the output of the following program:

```
// The default block is placed above  
other cases.
```

```
#include <stdio.h>
```

```
int main() {
```

```
    int x = 4;
```

```
    switch (x) {
```

```
        default: printf("Choice other than 1  
and 2");
```

```
            break;
```

```
        case 1: printf("Choice is 1");
```

```
            break;
```

```
        case 2: printf("Choice is 2");
```

```
            break;
```

```
    }
```

```
        return 0;
```

```
}
```

Output:

Choice other than 1 and 2

What would be the output of the following program:

```
void main( ) {  
    int c = 3 ;  
    switch ( c ) {  
        case 'v' :  
            printf ( "I am in case v \n" ) ;  
            break ;  
        case 3 :  
            printf ( "I am in case 3 \n" ) ;  
            break ;  
        case 12 :  
            printf ( "I am in case 12 \n" ) ;  
            break ;  
        default :  
            printf ( "I am in default \n" ) ;  
    }  
}
```

What would be the output of the following program:

```
void main( ) {  
    int i = 0 ;  
    switch ( i ) {  
        case 0 :  
            printf ( "\nCustomers are dicey" ) ;  
        case 1 :  
            printf ( "\nMarkets are pricey" ) ;  
        case 2 :  
            printf ( "\nInvestors are moody" ) ;  
        case 3 :  
            printf ( "\nAt least employees are good" ) ;  
    }  
}
```

What would be the output of the following program:

```
void main( ) {  
    int k, j = 2 ;  
    switch ( k = j + 1 ) {  
        case 0 :  
            printf ( "\nTailor" ) ;  
        case 1 :  
            printf ( "\nTutor" ) ;  
        case 2 :  
            printf ( "\nTramp" ) ;  
        default :  
            printf ( "\n Simple !" ) ;  
    } //end of switch  
} //end of main
```

What would be the output of the following program:

```
void main( ) {  
    int i = 1 ;  
    int ch = 'a' + 'b' ;  
        switch ( ch )          {  
            case 'a' :  
            case 'b' :  
                printf ( "\nYou entered b" ) ;  
            case 'A' :  
                printf ( "\na as in apple" ) ;  
            case 'b' + 'a' :  
                printf ( "\nYou entered a and b" ) ;  
        }  
}
```