Switch Case Statements

The switch statement in C is an alternate to if-else-if ladder statement which allows us to execute multiple operations for the different possibles values of a single variable called switch variable.

A **switch** statement allows a variable to be tested for equality against a list of values. Each value is called a case, and the variable being switched on is checked for each **switch case**.

The syntax of switch statement in c language

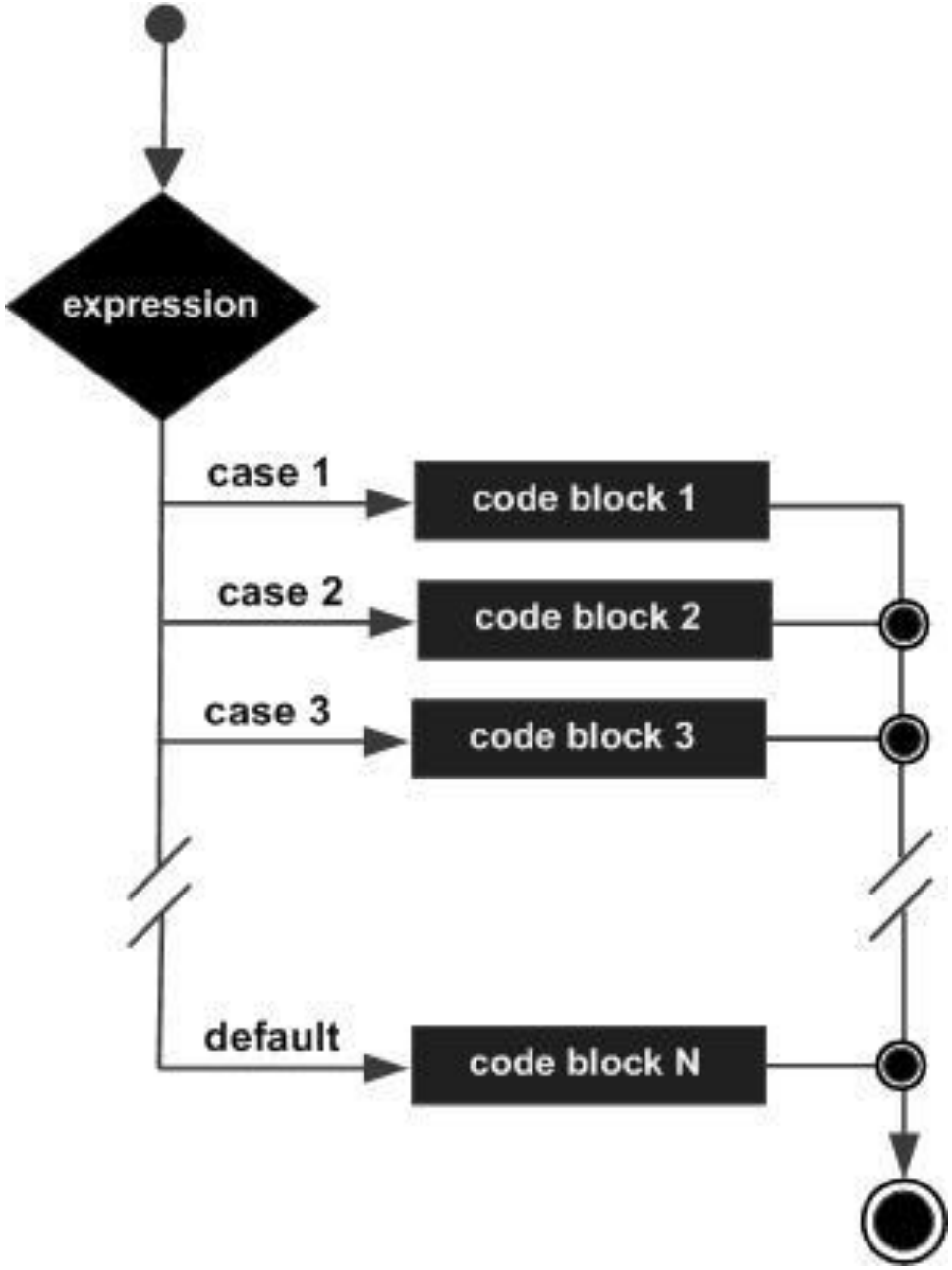
is given below:

```
switch(expression)
{
case value1:
        //code to be executed;
        break;  //optional
case value2:
        //code to be executed;
        break;  //optional
......
default:
//code to be executed if all cases are not matched;
}
```

The following rules apply to a switch statement −

1. The switch expression must be of an integer or character type.
2. 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.
3. When the variable being switched on is equal to a case, the statements following that case will execute until a break statement is reached.

4. The break statement in switch case is not must. It is optional. If there is no break statement found in the case, all the cases will be executed present after the matched case. It is known as fall through the state of C switch statement.
5. When a break statement is reached, the switch terminates, and the flow of control jumps to the next line following the switch statement.
6. A switch statement can have an optional default case, which must appear at the end of the switch. 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.



Example

```
#include <stdio.h>

int main () {

   /* local variable definition */
   char grade = 'B';

   switch(grade) {
      case 'A' :
         printf("Excellent!\n" );
         break;
      case 'B' :
      case 'C' :
         printf("Well done\n" );
         break;
      case 'D' :
         printf("You passed\n" );
         break;
      case 'F' :
         printf("Better try again\n" );
         break;
      default :
         printf("Invalid grade\n" );
   }

   printf("Your grade is  %c\n", grade );

   return 0;
}
```

When the above code is compiled and executed, it produces the following result

```
Well done
Your grade is B
```

Switch case example 2

```
#include <stdio.h>
int main()
{
   int x = 10, y = 5;
   switch(x>y && x+y>0)
   {
      case 1:
      printf("hi");
      break;
      case 0:
      printf("bye");
      break;
      default:
      printf(" Hello bye ");
   }

}
```

Output

```
hi
```

C Switch statement is fall-through

In C language, the switch statement is fall through; it means if you don't use a break statement in the switch case, all the cases after the matching case will be executed.

Let's try to understand the fall through state of switch statement by the example given below.

```
1. #include<stdio.h>
2. int main(){
3. int number=0;
4.
5. printf("enter a number:");
6. scanf("%d",&number);
7.
8. switch(number){
9. case 10:
10.     printf("number is equal to 10\n");
11.     case 50:
12.     printf("number is equal to 50\n");
13.     case 100:
14.     printf("number is equal to 100\n");
15.     default:
16.     printf("number is not equal to 10, 50 or 100");
17.     }
18.     return 0;
19.     }
```

Output

```
enter a number:10
number is equal to 10
number is equal to 50
number is equal to 100
number is not equal to 10, 50 or 100
```

Nested switch case statement

We can use as many switch statement as we want inside a switch statement. Such type of statements is called nested switch case statements. Consider the following example.

```
1. #include <stdio.h>
2. int main () {
3.
4.    int i = 10;
5.    int j = 20;
6.
7.    switch(i) {
8.
9.       case 10:
10.                printf("the value of i evaluated in outer switch: %d\n"
    ,i);
11.             case 20:
12.                switch(j) {
13.                   case 20:
14.                      printf("The value of j evaluated in nested switch:
    %d\n",j);
15.                }
16.          }
17.
18.          printf("Exact value of i is : %d\n", i );
19.          printf("Exact value of j is : %d\n", j );
20.
21.          return 0;
22.       }
```

Output

```
the value of i evaluated in outer switch: 10
The value of j evaluated in nested switch: 20
Exact value of i is : 10
Exact value of j is : 20
```

Practice Problems:

1. **Write a Menu Driven Program to display the month name by accepting the month number from the user.**

2. **Write a Program to display the class according to the marks scored by the students.**

   The Marks scored is taken as input and the class is displayed according to the following range.

   70-100 Distinction
   60-69 First Class
   50-59 Second Class
   40-49 Pass Class
   0-39 Fail

3. **Write a Menu Driven Program to perform add/subtract/multiply/divide/modulus based on the user's choice.**