Switch Case Statement:

we are going to learn about control statements. C language provides us a special control statement that allows us to handle different cases effectively instead of using a series of if statements.

The control statement that allows us to make a decision effectively from the number of choices is called a **switch**, or a **switch case-default** since these three keywords go together to make up the control statement. The expression in switch returns an integral value, which is then compared with the different cases. Switch executes that block of code, which matches the case value. If the value does not match with any of the cases, then **the default** block is executed. The syntax of the switch statement is:

```
switch ( integer expression )
{
case value 1 :
do this ;

case value 2 :
 do this ;

default :
do this ;
 }
```

Explanation of the general form of Switch Statement:-

The expression following the switch can be an integer expression or a character expression. The case value 1, 2 are case labels that are used to identify each case individually. Remember that case labels should be different. If it is the same, it may create a problem while executing a program. At the end of the Case labels, we always use a colon (:). Each case is associated with a block. A **block** contains multiple statements that are grouped for a particular case.

Whenever the switch is executed, the value of test-expression is compared with all the cases present in switch statements. When the case is found, the block of statements associated with that particular case will execute. The break keyword indicates the end of a particular case. If we do not put the break in each case, then even though the specific case is executed, C's switch will continue to execute all the cases until the end is reached. The default case is optional. Whenever the expression's value is not matched with any of the cases inside the switch, then the default will be executed.

Example of Switch in C :-

```c
#include <stdio.h>

    int main() {
        int i = 9;

        switch (i) {
            case 5:
            printf("Value is 7");
                break;

            case 0:
            printf("Value is 8");
                break;

            case 9:
            printf("Value is 9");
                break;

            default:
            printf("Value is not present");
                break;
        }
        return 0;
    }
```

Note -: It is not necessary to use break keyword after every case. When we do not want to terminate our case at that time, we will not use the break keyword.

Nested Switch in C:

We can also use nested switch statements i.e., switch inside another switch. Also, the case constants of the inner and outer switch may have common values without any conflicts.

The syntax of the nested switch is:

```
switch(expression 1) {
Case 1:
  printf(“Switch Statement 1”);

switch(expression 2) {
Case 1:
  printf(“Switch Statement 2”);
  break;

Case 2:
Do this;
}
  break;

Case 2:
Do this;
}
```

Why do we need a Switch case?

There is one problem with the if statement: the program's complexity increases whenever the number of if statements increases. If we use multiple if-else statements in the program, the code might become difficult to read and comprehend. Sometimes it also even confuses the developer whom himself wrote the program. Using the switch statement is the solution of this problem.

Rules for Switch Statement -:

1. **The test expression of** Switch must be an int or char.
2. **The value of the** case should be an integer or character.
3. Cases should be inside the switch statement.
4. Using the break keyword in the switch statement is not necessary.
5. The case label values inside the switch should be unique.

Difference between a switch and if:

- Switch statements cannot evaluate float conditions, and the test expression can be an integer or a character, whereas if statements can evaluate float conditions.
- The switch statement cannot evaluate relational operators i.e., they are not allowed in switch statements, whereas if statements can evaluate relational operators. switch
- Cases in the switch can never have variable expressions; for example, we cannot write case a +3 :