DECISION MAKING IN PROGRAMMING USING SWITCH OPERATOR

Programming using the decision statements, it is necessary to use the most appropriate decision statement according to the problem. 'If' statement is good for choosing between two alternatives, it quickly becomes cumbersome when several alternatives are needed. C++ language's solution to this problem is the 'switch' statement.

Switch Statement

The 'switch' statement is C++'s multiple selection statement. It is use to select one of several alternative paths in program execution.

Working of Switch Statement

A variable is successfully tested against a list of integer or character constants. When a match is found, the statement sequence associated with that match is executed.

General Form

The general form of the 'switch' statement is as follows:

```
switch (variable)
{
case constant 1:
statement sequence
break;

case constant 2:
statement sequence
break;

case constant 3:
statement sequence
break;
.
.
default:
statement sequence;
}
```

Where the default sequence is performed if no matches is found. The default is optional. If all matches fail and default is absent, no action takes place. When match is found, then the statement associated with the case is executed until the break statement is encountered or the end of switch statement is reached.

Comparison of Switch with If Statement

The ‘switch’ statement differ from the ‘if’ statement, in that ‘switch’ can only test for equality, whereas the ‘if’ condition is of any type. Also, ‘switch’ will only work with integer or character types.

Break statement

The ‘break’ statement allows exiting a loop from any point with in its body, by passing its normal termination expression. When the ‘break’ statement s encountered inside a loop, the loop is immediately terminated and programs control resume at the next statement following the loop. The ‘break’ statement is commonly used in loops in which a special condition can cause immediate termination.