Control Statements

By default, the statements in the program are executed sequentially.

The control statements alter the execution of statements depending upon the conditions specified inside the parenthesis.

Which is used to make a decision for real world problems

Types

a. Simple if
b. if .. else
c. Multiple else if or else if ladder
d. Nested if
e. switch
f. break or continue
g. goto statement

(a) Simple if

Used to Evaluate if block statements if its is true, otherwise skip the if block

Syntax:

```
if(condition)
{
        Execute block of statements;
        ----------------;
        ----------------;
 }
Normal statements;
```

(b)If .. Else Statement

Used to Evaluate True or False

If the if condition is true it will execute true block otherwise, execute else block statements

Syntax:

```
if(condition)
{
        Execute true block statements;
        ----------------;
        ----------------;
}
else
{
```

Else block statements;

}

(c) Multiple else if Statements

Used to evaluate Series of Conditions

If we have to use more than one if-else statement, we have to create Multiple else if statements.

Syntax:

if(condition 1)

{

Execute 1st true block statements;

}

else if(condition 2)

{

Execute 2nd block statements;

}

(d) Nested If statements

One condition followed by another condition

Syntax:

```
if(condition1)
{
        if(condition2)
        {
                Inner if statements;
        }
        else
        {
                Inner else statements;
        }
}
else
{
Outer else statement;
}
```

- **Note for else statement**
    i. Else should be followed by if statement

ii. Else should not take condition
iii. Else will execute only if the immediate if statement false
iv. Else is optional

(e)Switch Statement

Switch statement allows us to make a decision from a number of choices.

Syntax:

```
switch(int var or char var)
{
        case constant1 : statement(s);
                        ----------;
        case constant2 : statement(s);
        case constant3 : statement(s);
        default: default statement(s);
}
```

- Points to be noted for switch statement
  (a) Only one variable can be tested with the available case statements
  (b) It should be either integer or character variable
  (c) Float, double, long type variables cannot be used as cases.
  (d) Default statement will execute only none other than case is match
  (e) Default statement may appear any where inside the switch statement
  (f) Default is optional

(f) The break Statement

Break statement is used to terminate or to exit from switch or inner loop.

Syntax:

```
{
        ----------;
        break;
        ----------;
        ----------;
}
```

(g)Goto statement

- Goto statement is used to transfer the control in a loop or a function from one point to any other portion in that program
- If misused the goto statement can make a program infinite or impossible to understand.

Syntax:

```
goto label;
Statement(s);
---------------;
--------------;
Label :
        statement(s);
```