Control Statements

Control statements are used to control the sequence of statements execution.

Classifications Control statements

1) Decision making statements:

1. If Statement
2. If Else Statement
3. Else if ladder
4. Nested If else

2) Selection Statements:

1. Switch Case

3) Looping Statements:

1. While
2. Do While
3. For Loop

4) Jump Statements:

1. break
2. Continue

1) Decision making statements:

Decision making statements in programming languages decide the direction of flow of program execution.

1.If Statement
2. If Else Statement
3.Else if ladder
4.Nested If else

- **If Statement**

if statement is the most simple decision making statement. It is used to decide whether a certain statement or block of statements will be executed or not i.e if a certain condition is true then a block of statement is executed otherwise not.

Syntax :

```
if(condition)
{
Statements 1;
}
```

- **If Else Statement:**

It include two parts:

1.True part

2.False part

Syntax:

```
if(condition)
{
 Statements 1;
}
Else
{
Statements 2;
}
```

- **Else if ladder:**

(One by one check)

1.The else if ladder is used to test set of condition in a sequence.
2.It is used when there are multiple condition to be check

Syntax:

```
if(condition)
{
 Statements 1;
}
else if (condition 2)
{
Statements 2;
}
else if (condition 3)
{
Statements 3;
}
      else
{
Statements n;
}
```

- **Nested If else:**

When if statement inside another if statement is know as nested if.

```
if(condition)
{
if(condition)
        {
         Statements 1;
        }
else
        {
        Statements 2;
        }
        }
else
        {
        Statements 3;
        }
```

2) Selection Statements:

- **Switch Case:**

Switch case statement is used when we have multiple condition and we need to perform different action based on the condition.

Syntax:

```
switch (expression)
{
 case constant1:
// statements
break;
case constant2:
// statements
 break; . . .
default:
// default statements
 }
```

3) Looping Statements:

1.While
2.Do While
3.For Loop

1.While Statement:

A while loop is a simple loop that will run the same code over and over as long as a given conditional is true. The condition is checked at the beginning of each run through the loop (including the first one). If the conditional is false for the beginning, the while loop will be skipped all together.

Syntax:

```
While(condition)
{
Statement 1;
}
```

2.Do While:

A do-while loop acts just like a while loop, except the condition is checked at the end of each pass through the loop body. This means a do-while loop will execute at least once.

syntax

```
do {
      // loop body
   } while ( condition );
```

3.For Loop

In for loop the initialization, condition ,or increment or decrement of loop variable is implemented in a single statement

Syntax:

```
for( initialization ; conditional ; iteration )
{
      // loop body
   }
```

4) Jump Statements:

1.break

The break statement can also be used to jump out of a loop

2.Continue

It is used to skipping for a particular part of loop