Branching:

Branching is so called because the program chooses to follow one branch or another.

if statement

This is the most simple form of the branching statements.

if the expression is true then the statement or block of statements gets executed otherwise these statements are skipped.

if statements take the following form:

Show Example

```
if (expression)
  statement;

or

if (expression)
  {
    Block of statements;
  }

or
if.......else

if (expression)
  {
    Block of statements;
  }
else
  {
    Block of statements;
  }

or
```

```
Nested if.......else

if (expression)
  {
    Block of statements;
  }
else if(expression)
  {
    Block of statements;
  }
else
  {
    Block of statements;
  }
```

switch statement:

The switch statement is much like a nested if .. else statement. Its mostly a matter of preference which you use, switch statement can be slightly more efficient and easier to read.

Show Example

```
switch( expression )
     {
        case constant-expression1:    statements1;
        [case constant-expression2:   statements2;]
        [case constant-expression3:   statements3;]
        [default : statements4;]
     }
```

Using break keyword:

If a condition is met in switch case then execution continues on into the next case clause also if it is not explicitly specified that the execution should exit the switch statement. This is achieved by using *break* keyword.

Try out given example Show Example

What is default condition:

If none of the listed conditions is met then default condition executed.

Try out given example Show Example

Looping

Loops provide a way to repeat commands and control how many times they are repeated. C provides a number of looping way.

while loop

The most basic loop in C is the while loop.A while statement is like a repeating if statement. Like an If statement, if the test condition is true: the statments get executed. The difference is that after the statements have been executed, the test condition is checked again. If it is still true the statements get executed again.This cycle repeats until the test condition evaluates to false.

Basic syntax of while loop is as follows:

Show Example

```
while ( expression )
{
   Single statement
   or
   Block of statements;
}
```

for loop

for loop is similar to while, it's just written differently. for statements are often used to proccess lists such a range of numbers:

Basic syntax of for loop is as follows:

Show Example

```
for( expression1; expression2; expression3)
{
   Single statement
   or
   Block of statements;
}
```

In the above syntax:

- expression1 - Initialisese variables.
- expression2 - Condtional expression, as long as this condition is true, loop will keep executing.
- expression3 - expression3 is the modifier which may be simple increment of a

variable.

do...while loop

do ... while is just like a while loop except that the test condition is checked at the end of the loop rather than the start. This has the effect that the content of the loop are always executed at least once.

Basic syntax of do...while loop is as follows:

Show Example

```
do
{
   Single statement
   or
   Block of statements;
}while(expression);
```

break and continue statements

C provides two commands to control how we loop:

- break -- exit form loop or switch.
- continue -- skip 1 iteration of loop.

You already have seen example of using break statement. Here is an example showing usage of **continue** statement.

```
#include

main()
{
    int i;
    int j = 10;

    for( i = 0; i <= j; i ++ )
    {
       if( i == 5 )
       {
          continue;
       }
       printf("Hello %d\n", i );
    }
}
```

This will produce following output:

```
Hello 0
Hello 1
Hello 2
Hello 3
Hello 4
Hello 6
Hello 7
Hello 8
Hello 9
Hello 10
```
