Statements: Switch statement, if statement, while statement, for statement

```
switch(expression)
        {
         case constant1:
                statements 1;
        break;
        case constant2:
                statements 2;
        break;
        ……..
        default:
                statements n;
        break;
        }
```

- if (expression)
  Statement;
- if(expression)
  statement1;
  else
  statement2;
- if(condition)
  Statement 1;
  else if (condition)
  Statement 2;
  .....
  .....
  else if(condition)
  Statement n-1;
  else
  Statements n;

```
for (initialization, condition, step)
                {
                 Statement 1;
                 Statement 2;
                 ……..
                 Statement n;
                }
```

```
while (expression)
      {
      Statement 1;
      Statement 2;
      ……..
      Statement n;
      }
```

do-while loop:

```
do
    {
    Statement 1;
    Statement 2;
    ……..
    Statement n;
    }
while(expression);
```


Goto:

```
#include  <stdio.h>
#include  <conio.h>
void main()
{
    int  number;
    clrscr();
    printf("www.");
    goto x;
    y:
    printf("expert");
    goto z;
    x:
    printf("cprogramming");
    goto y;
    z:
    printf(".com");
    getch();
}
```

```
www.cprogrammingexpert
```

Null statement:

```
#include <stdio.h>
void main()
{
    int i;
    clrscr();
    for(i=1;i<=5;i++)
        {
        if(i%2==0)
            ;     //null statement
        else
            printf("%d\n",i);

        }
    getch();
    return 0;
}
```

```
1
3
```

Continue statement:

```
#include<conio.h>
#include<stdio.h>
int main()
{
    int     i;
    clrscr();
    for(i=1;i<=5;i=i+1)
        {
        if(i==3)
            continue;
        printf("%d\n",i);
        }
    getch();
    return 0;
}
```

```
1
2
4
```

Break statement:

The break statement is a jump instruction and can be used inside a switch construct, for loop, while loop and do-while loop.

Operators:

Arithmetic operators:

Operator	Meaning
+	Addition or Unary plus
-	Subtraction or Unary minus
*	Multiplication
/	Division
%	modulo Division

Relational operators:

Operator	Meaning
<	Less than
>	Greater than
<=	Less than or equal to
>=	Greater than or equal to
==	Equal to
!=	Not equal to

Logical operators:

Operator	Meaning
&&	Logical AND
||	Logical OR
!	Logical NOT

Assignment operator:

Operator	Equation	Output
+=	i=i+1	i+=1
-=	i=i-10	i-=10
*=	i=i*11	i*=11
/=	i=i/12	i/=12
%=	i=i%5	i%=5

Conditional operator:

The conditional expression can be used as shorthand for some if-else statements. It is a ternary operator. This operator consist of two symbols: the question mark (?) and the colon (:).

Identifier = (test expression)? Expression1: Expression2;