

TOKENS

- A token is source-program text that the compiler does not break down into component elements
- The keywords, identifiers, constants, variables, and operators described in this section are examples of tokens. Punctuation characters such as brackets ([]), braces { }, parentheses (()), and commas (,) are also tokens.



TOKEN

- **Keywords:** keywords are the reserved words with specific meaning or task there are 48 keywords in C++.
- **Identifiers:** It is the name given to a variable, function, array etc. These are the user defined names and consist of the sequence of the letters and digits with a letter as a first character.



CONSTANTS

- **Integer constant**-Consists of the digits.
- **Float constant**-Consists of the digits with decimal.
- **Character constant**-single character enclosed with in a pair of single quote.
- **String constant**-sequence of characters enclosed in double quotes



OPERATORS

- A operator is a symbol that tells the computer to perform certain mathematical or logical operations.
- Two types
 - Unary operator
 - Binary operator



UNARY OPERATOR

- Unary operator have only one operand.
- It is of 3 types:
 - Unary minus
 - Increment.
 - Post increment
 - Pre increment
 - Decrement:
 - Post decrement
 - Pre decrement



BINARY OPERATOR

- Binary operator have two operands.
- It is of five types:
 - Arithmetic operators
 - Relation operators
 - Logical operators
 - Assignment operator
 - Conditional operator



CONTROL STATEMENTS

- Control Statements are elements in source code that control the flow of program execution. There are blocks using { and }, loops using conditions, switch statements, loops and jumping statements.



CONDITIONAL STATEMENTS

The if statement: A conditional statement decides whether to execute code based on conditions. It works on one or more than two conditions and selects one option

- It is of three types:
 - Simple if-else
 - leader if-else
 - nested if-else



Simple if else

- Syntax:

```
if(condition)
```

```
{
```

```
True statements
```

```
}
```

```
else
```

```
{
```

```
False statements
```

```
}
```



SWITCH STATEMENT

- The switch statement is the multi branching statement. Switch statement is used when there is a possibility to make a choice from a number of options.



SYNTAX

```
Switch(expression)
```

```
{
```

```
case 1:
```

```
{
```

```
}
```

```
Case n:
```

```
{
```

```
}
```

```
default:
```

```
{
```

```
}
```

```
}
```



ITERATION STATEMENTS

- An iteration statement creates a *loop* of code to execute.
- A looping statement is of three types:
 - The for loop
 - The do...while loop
 - The while loop



FOR LOOP/DO WHILE

- For loop:

```
for( initialization; condition; inc/dec)
{
  Statements
}
```

- do..while loop

```
initialization
do
{
  Statements
  Inc/dec;
}while(condition);
```



WHILE LOOP

- Syntax

Initialization;

while(condition)

{

Statements

Inc/dec;

}



JUMP STATEMENTS

- A jump statement can be used to transfer program control using keywords such as **break**, **continue**, **return**, **yield**, and **throw**.
 - **break**
 - **continue**
 - **return**

