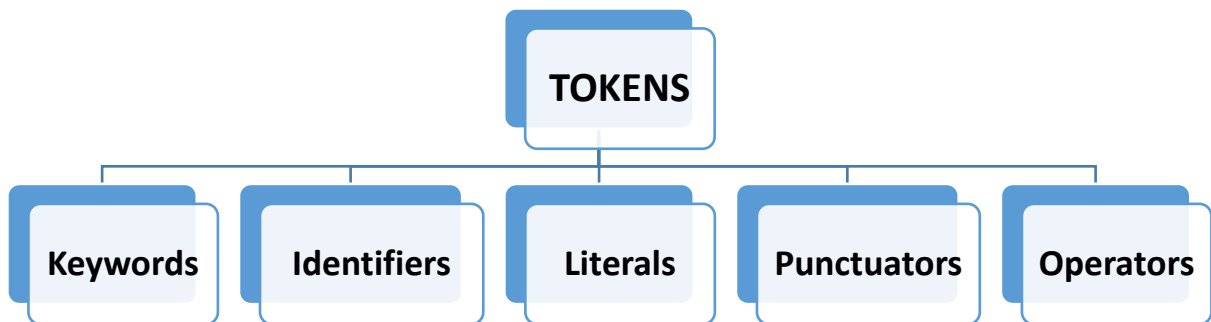


HOLY ANGEL SR. SEC. SCHOOL

CLASS : IX

Sub: Computer Applications

JAVA TOKENS



1. Keywords : These are the words which convey special meaning to the language compiler. For e.g. class, if, private, main etc.

Note: true, false and null are not keywords but reserved words.

2. Identifier : Name given to variables, objects, class, function, arrays etc. is called Identifier.

Identifier Naming Conventions are:

- a) They can have alphabets, digits, underscore and dollar sign characters.
- b) They can not be a keyword or reserved word.
- c) They must not begin with a digit.
- d) They can be of any length.
- e) Java is case sensitive.

Following are some **valid identifiers**:

Myfile date_of_birth _CHK \$1 etc.

Invalid Identifiers:

Data-Rec 3Miles main My.weight

3. Literals or constants : Literals are data items that have fixed values. They are of many types:
 - (i) Integer – literal : for e.g. 345, -908, 6535 etc.
 - (ii) Real or floating Literals : for e.g. 5.0, -42.7, 678.90 etc.
 - (iii) Boolean literal : true and false
 - (iv) Character Literal: they are enclosed in single quotes. e.g. 'z', 'A' etc.
 - (v) String Literal : Multiple character constant e.g "Hello India".
 - (vi) The Null Literal : It is of null type.
4. Operators : They are of many kinds.
 - a. Mathematical : +, -, /, *, %
 - b. Relational : <, <=, >, >=, ==, !=
 - c. Logical : &&(AND), ||(OR), !(NOT)
 - d. Conditional : (expression) ? True value : False value;
5. Separators (Punctuators): () { } [] ; , .

DATA TYPES

Data can be of many types e.g. character, integer, real, string etc.

Data Types are means to identify the type of data and associated operations of handling it.

Java Data Types are of two types:

Primitive or Fundamental Data Types : byte, short, int , long, float, double, char, Boolean.

Reference Data Types : They are constructed from primitive data types. For e.g. classes, arrays and interface.

Primitive DataTypes:

(a) Numeric Integral Types

Type	Size(in bytes)	Range
------	----------------	-------

byte	1 byte	- 128 to +127
short	2 bytes	- 32,768 to +32,767
int	4 bytes	-2 billion to +2 billion (approx..)
long	8 bytes	-2 ⁶³ to +2 ⁶³ -1 (approx..)

(b) Fractional Numeric Types

Type	Size	
float	4 bytes	Precision upto 6 decimal places
double	8 bytes	Precision upto 15 decimal places

Java assumes the fractional numbers to be of double data type unless specified. To explicitly specify their type, use suffixes, F or D.

(c) Character Type

char	2 bytes	0 to 65,535	Unicode Characters
------	---------	-------------	--------------------

(d) Boolean Types

boolean	1 byte	true or false
---------	--------	---------------

Variables:

A variable is a named memory location, which holds a data value of a particular data type.

Declaration : e.g. int a;

Initialization : a=20;

Write a program of all data types:

```
public class example
{
    public static void main(String args[])
    {
        byte a=5;
        short b=1287;
        int c=1675;
        float d=3.14f;
    }
}
```

```
long e=3456;
double f=22.75;
char g='h';
string h="Java";
boolean i=true;
```

```
System.out.println("Byte value =" +a);
System.out.println("Short value=" +b);
```

```
.
.
.
.
.
..
}
}
```

Operators in JAVA

1. Arithmetic Operators

2 types: Unary and Binary

Unary (+ and -):

If a = 5 then +a means 5.

If a = - 4 then +a means -4.

If a = - 6 then -a means 6.

Binary (+, -, *, /, %) – work on two operands.

For e.g. 4 + 20 results in 24

30 / 6 results in 5

3 * 4 results in 12

20 % 3 results in 2 (Modulus operator gives remainder)

2. + Operator with Strings

```
int a=5, b=7
```

```
System.out.println("My first file"); // output : My first file.
```

```
System.out.println("My first file"+a); // output : My first file 5
```

```
System.out.println("a+b"); //output : a+b
```

```
System.out.println(a+b); // output : 12
```

```
System.out.println("Addition"+(a+b)); //output: Addition 12
```

```
System.out.println("Addition"+a+b); //output : Addition 57
```

3. Increment and Decrement Operator (++ , --)

They are of 2 types:

Pre-increment

```
int a=5, sum=0;
```

```
sum=sum + (++a);
```

```
// sum will be 6 and a also 6
```

Post – increment

```
int a=5, sum=0;
```

```
sum=sum + a++;
```

```
// sum will be 5 and a  
will be 6.
```

Pre-decrement

```
int a=5, sum=0;
```

```
sum= sum + (--a);
```

```
// sum and a both will be 4.
```

Post decrement

```
int a=5, sum=0;
```

```
sum= sum + a--;
```

```
// sum will be 5 and a will be 4.
```

4. Relational Operators

< less than

3<5, 4<6

<= less than equal to 3<=5, 3<=3
> greater than 7>1, 1>3
>= greater than equal to 7>=3, 7>=9
== equal to 3==3
!= not equal to 3!=1, 5!=7

Use of Relational Operators

```
public class relational
{
    public static void main(String args[])
    { int a=5, b=7;
        System.out.println(5<b);
        System.out.println(11<=16);
        System.out.println(a>b);
        System.out.println(a>=5);
        System.out.println(a==6);
        System.out.println(b==7);
        System.out.println(b!=4);
        System.out.println(a!=7);
        boolean c=a<b;
        System.out.println(c);
    }
}
```

5. Logical Operators

Logical AND (*) && 3<5 && 2<4

Logical OR (+) || 3<5 || 4<8

Logical NOT (-) ! !(3<5)

Write a program to use Logical Operator

```
public class logical_opt
{
    public static void main(String args[])
    {
        int a=10, b=15;
        System.out.println((a>15) && (b<=10));
        System.out.println((a==15) || (b!=10));
        System.out.println(!(a>5));
    }
}
```

Operator Precedence and Associativity

Operator	Remark	Associativity
[], ()	(used to group expressions. [] is used for arrays.	Left to right
++, --, !		Right to left
*, /, %	Multiplication, division, modulus	Left to right
+, -	Addition, subtraction	Right to left
<, >, <=, >=	Relational comparison test	Left to right
==, !=	Equality and not equal to	Left to right
&&	Logical AND	Left to right
	Logical OR	Left to right
=, +=, -=, *=, /=, %=, ^=	Assignment and shorthand assignments	Right to left