**Java**

**Topics:**

1.Introduction

2.java comments

3.DataTypes

4.Variables

5.Operators

6.control structures

**1.Introduction**

Java is a popular programming language, created in 1995.

It is owned by Oracle, and more than **3 billion** devices run Java

Java is a **multi-platform**, **object-oriented**, and **network-centric language** that can be used as a platform in itself

**Multi-platform**: Java is designed to be platform-independent, meaning that Java programs can run on any platform that has a compatible Java Virtual Machine (JVM). This allows developers to write code once and run it on different operating systems, such as Windows, macOS, Linux, or even embedded systems.

**Object-oriented**: Java follows the principles of object-oriented programming (OOP), which organizes code around objects that encapsulate data and behavior.

**Network-centric**: Java has extensive built-in libraries and features that facilitate network programming. It provides classes and APIs (Application Programming Interfaces) for networking, such as socket programming, remote method invocation (RMI), and various protocols like HTTP and FTP

**2.java comments**

They are two types of comments:

1)Single line

2) Multi-line

**2.1)Single line**

Single-line comments start with two forward slashes (//).

Any text between // and the end of the line is ignored by Java (will not be executed).

**Example:**

Class Demo

{

Public static void main (String[]args)

{

**// This is a comment**

System.out.println("Hello World");

}

}

**2.2)Multi-line**

Multi-line comments start with /\* and ends with \*/.

Any text between /\* and \*/ will be ignored by Java.

Example:

Class Demo

{

Public static void main (String[]args)

{

**/\*this java comments**

**Java program multi line comments\*/**

System.out.println("Hello World");

}

}

**3.DataTypes:**

Data: Data means collection of information.

Data types are used to indicate the type of data stored into a variable

Data types are divided into two groups:

3.1.Primitive data types - includes byte, short, int, long, float, double, boolean and char

3.2Non-primitive data types - such as [String](https://www.w3schools.com/java/java_strings.asp), [Arrays](https://www.w3schools.com/java/java_arrays.asp) and [Classes](https://www.w3schools.com/java/java_classes.asp) (you will learn more about these in a later chapter)

**4.Variables:**

Variables are containers in order to store some data or information.

Ex:

Age(variable)=20(value)

Height=5.8

**4.1.Variable Declaration**

**Syntax:** Datatype variableName;

Ex:

Int age

Double height

**4.2.variable initialization**

**Syntax:** variableName=value;

Ex:

Age=25;

Height=6.5;

**4.3.variable Declaration & initialization**

**Syntax:** Datatype VariableName=value;

Ex:

Int marks=99;

String name=”sinchan”;

**Programs:**

**public** **class** Student {

**public** **static** **void** main(String[] args) {

**int** age;//variable declaration

age=20;//variable initialization

String name="sinchan";//declaration & initilization

System.***out***.println(age);

System.***out***.println(name);

}

}

### Types of Variables

There are three types of variables in java:

* **local variable**

A variable declared inside the body of the method is called local variable. You can use this variable only within that method and the other methods in the class aren't even aware that the variable exists.

* **instance variable**

A variable declared inside the class but outside the body of the method, is called an instance variable.

* **static variable**

A variable that is declared as static is called a static variable. It cannot be local. You can create a single copy of the static variable and share it among all the instances of the class

**Ex:**

Public class A

**static** **int** m=100;//static variable

**void** method()

    {

**int** n=90;//local variable

    }

**public** **static** **void** main(String args[])

    {

**int** data=50;//instance variable

    }

}//end of class

**5.Operators**

Operators are used to perform operations on variables and values.

* Unary Operator,
* Arithmetic Operator,
* Shift Operator,
* Relational Operator,
* Bitwise Operator,
* Logical Operator,
* Ternary Operator and
* Assignment Operator.

**Unary Operator**

Unary operators need only one operand. They are used to increment, decrement, or negate a value.

**++ increment** (increment by 1)

**-- decrement**(decrement by 1)

**5.1.1.++ increment**

Post increment pre increment

**Rules: Rules:**

1)First assign the values 1)First increment the value

2)then increment 2)then assign

Ex: Ex:

X=5; x=6 x=3; x=3

Y=x++; y=5 y=++x; y=4

**Program:**

Class Demo

{

**public** **static** **void** main(String[] args) same code copy

{

**int** x = 10;

**int** y=x++;

System.***out***.println(x);

System.***out***.println(y);

}

}

**-- decrement**

Post decrement pre decrement

**Rules: Rules:**

1)First assign the values 1)First decrement the value

2)then decrement 2)then assign

**Logical Operator**

Logical operators are used to determine the logic between variables or values.It always return Boolean values.

**&& -** Logical and

**|| -** Logical or

**! -** Logical not

**Example:**

1.To login a Email account you should have user id **&&(logical and)** password.(Account will login only if both user id and password are true).

2.your email account will be logged in either one of your mobile number **||(logical or)** your email address is True

3)! Not means negation(by using truth tables)

**Ternary Operator**

Java Ternary operator is used as one line replacement for if-then-else statement and used a lot in Java programming. It is the only conditional operator which takes three operands.

**Syntax**:

Condition ? true :false;

**Example:**

Class Demo

{

Public static void main(String [] args)

{

Int a=5;

Int b=10;

System.out.println((n%2)==0 ? ”even” : ”odd” );

System.out.println((n%2)==0 ? ”even” : ”odd” );

**}**

**}**

**6.control structures**