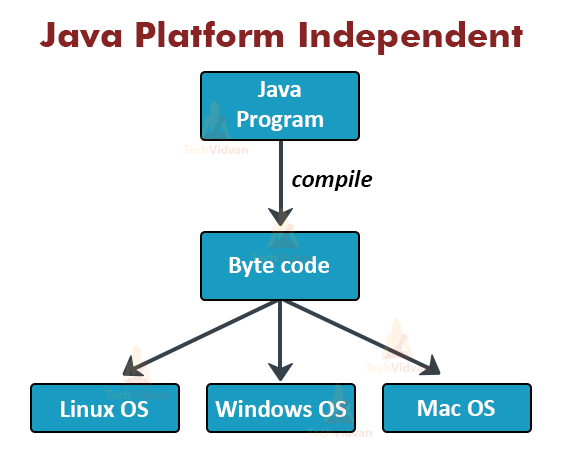
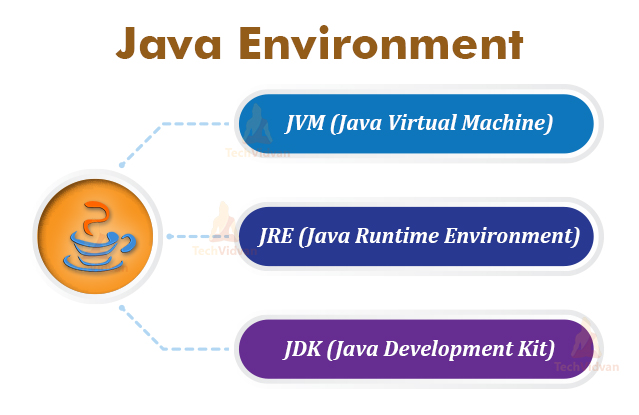
### Introduction to Java

Java programming language is a high-level, object-oriented, general-purpose, and secure programming language. Java is the most widely used programming language. It is designed for the distributed environment of the Internet. Java is freely accessible to users, and we can run it on all the platforms. Java follows the WORA (Write Once, Run Anywhere) principle, and is platform-independent.



### Java Environment- JVM, JRE, and JDK

[](https://i0.wp.com/techvidvan.com/tutorials/wp-content/uploads/sites/2/2020/06/Java-Environment-tv.jpg?ssl=1)

**Features of Java**

**1. Simple:** Java is simple because its syntax is simple and easy to understand.

**2. Object-Oriented:** Everything in Java is in the form of the object. In other words, it has some data and behaviour. A Java program must have at least one class and object.

**3. Robust:** Java always tries to check errors at runtime and compile time. Java uses a garbage collector to provide a strong memory management system. Features like Exception handling and garbage collection make Java robust or strong.

**4. Secure:** Java is a secure language, as Java does not use explicit pointers. All Java programs run in the virtual machine. Moreover, Java contains a security manager that defines the access levels of Java classes.

**5. Platform-Independent:** Java provides a guarantee that to write code once and run it anywhere(at any platform). The compiled byte code is platform-independent, and we can run it on any machine irrespective of the Operating system.

**6. Portable:** We can carry the byte code of Java to any platform. There are no implementation-dependent features in Java. Java provides predefined information for everything related to storage, such as the size of primitive data types.

**7. Multi-threaded:** Java is a multi-threaded language as it can handle more than one job at a time.

#### 1. JVM (Java Virtual Machine)

**Interpreter to execute java programs (Byte code) line by line**

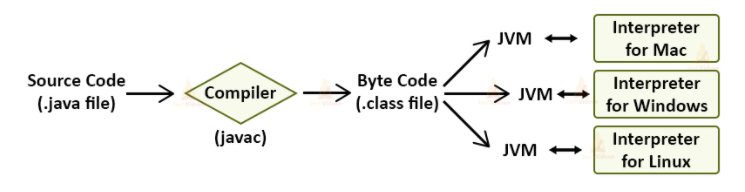
* Loading the code
* Verifying the code
* Executing the code
* Providing a runtime environment

#### 2. JRE (Java Runtime Environment)

**Environment to run java applications**

#### 3. JDK (Java Development Kit)

**Environment to develop, compile and run java application**



### Java Variables

Computer programs read data from input devices like keyboard, mouse etc. They process this input data and write it to an output device or network. Java stores the program data in variables.

Java program first declares the variables, reads data into these variables, executes operations on the variables, and then writes them somewhere again.

There are the following types of variables in Java basics:

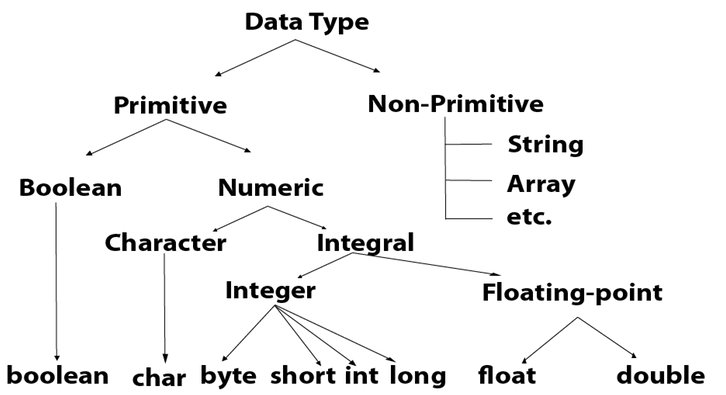
1. Local Variables
   * Which are accessed with the particular method and it will be local to that method
2. Class Variables (Static Variables)
   * Static variable will get loaded with the class loading
3. Instance Variables (Non-static Variables)
   * Instance variables are those which is related to class object

### Java Data Types

There is a basic java data type for each variable in Java. The data type of a variable determines the type of data the variable can contain, and what operations we can execute on it.

Every bit of processed data every day is divided into types. The type of data is called a data type. There are various kinds of data types in Java.

Broadly, the data types are mainly of two categories:

******

#### a. Primitive Data Types in Java

Primitive data types are fundamental data types offered by Java. These are the basic data values. These data types are hard coded into the Java compiler so that it can recognize them during the execution of the program.

There are 8 types of primitive data types in Java:  
a. int  
b. float  
c. char  
d. boolean  
e. byte  
f. short  
g. long  
h. double

#### b. Non-Primitive Data Types in Java

Non-Primitive data types are the reference data types. These are the special data types that are user-defined. The program already contains their definition. Some examples of non-primitive or reference data types are

1. Classes
2. Interfaces
3. String
4. arrays, etc.

### Java Operators

Java Operators are the special type of tokens. When they are coupled with entities such as variables or constants, they result in a specific operation. The operation can be any, such as addition, multiplication or even shifting of bits, etc.

There are the following types of Java operators;

* Arithmetic Operators
* Logical Operators
* Comparison Operators
* Assignment Operators

### Java Method

A method or function basically defines a behaviour. There can be a number of methods in Java. In methods, there are logics written. We can manipulate data in methods and also execute actions on them.

### Comments in Java

Comments are needed whenever the developer needs to add documentation about a function that is defined within the program. This is to enhance code readability and understandability. Comments are not executed by the compiler and simply ignored during execution.

The comments are of the following types:

#### a. Single-Line Comments in Java

The single-line comments consist of a single line of comment. We generally write them after a code line to explain its meaning. We mark the single-line comments with two backslashes(//).

String str = “DataFlair”;  
// The value of i is set to 6 initially. The string has value “DataFlair”

#### b. Multi-Line Comments in Java

Multi-line comments, as the name suggests, span for multiple lines throughout the code. We generally write to them at the beginning of the program to elaborate on the program.

Developers also use them to comment out blocks of code during debugging. We mark them using starting tag(/\*) and an ending tag(\*/).

### Java Class

Class in Java is the blueprint that defines similar types of objects derived from it. A class represents a set of methods and properties that are common to all the objects.

A class is one of the fundamental building blocks of Object-Oriented programming in Java. We can define a class in Java using the class keyword.

### Java Object

An object in java is an identifiable entity that has some characteristics and behaviour. We create objects from class in Java.

For example, a Fan is an object that has three characteristics: It has three blades, It has a brown color, etc. Its behaviour is: it rotates at some speed. We can create the object of a class once we define a class.

### Java Constructors

A Constructor in Java is a block of code that creates an object. Therefore, it is also called an object builder in Java. The constructor is very similar to a Java method. The main difference between them is that a constructor does not have a return type, not even void, unlike methods.

We can often call it a method. The constructor automatically gets invoked during the object created using a new operator. The constructor’s name should be the same as the class name.

There are two types of constructors in Java:

1. Default constructor
2. Parameterized constructor

|  |  |  |
| --- | --- | --- |
| **S.N.** | **Type Casting** | **Type Conversion** |
| 1 | Type casting is a mechanism in which one data type is converted to another data type using a casting () operator by a programmer. | Type conversion allows a compiler to convert one data type to another data type at the compile time of a program or code. |
| 2 | It can be used both compatible data type and incompatible data type. | Type conversion is only used with compatible data types, and hence it does not require any casting operator. |
| 3 | It requires a programmer to manually casting one data into another type. | It does not require any programmer intervention to convert one data type to another because the compiler automatically compiles it at the run time of a program. |
| 4 | It is used while designing a program by the programmer. | It is used or take place at the compile time of a program. |
| 5 | When casting one data type to another, the destination data type must be smaller than the source data. | When converting one data type to another, the destination type should be greater than the source data type. |
| 6 | It is also known as narrowing conversion because one larger data type converts to a smaller data type. | It is also known as widening conversion because one smaller data type converts to a larger data type. |
| 7 | It is more reliable and efficient. | It is less efficient and less reliable. |
| 8 | There is a possibility of data or information being lost in type casting. | In type conversion, data is unlikely to be lost when converting from a small to a large data type. |
| 8 | float b = 3.0;  int a = (int) b | int x = 5, y = 2, c;  float q = 12.5, p;  p = q/x; |

### Conditional Statements in Java

### \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Conditional statements are statements that are purely based on the condition flow of the program. There are mainly three types of conditional statements in Java:

#### i. Java if statement

The if statement suggests that if a particular statement results in true then the block enclosed within the if statement gets executed.

#### ii. Java if-else statement

The if-else statement states that if a particular condition is true then the if block gets executed. If the condition is false, then the else block gets executed.

#### iii. Java Else if statement

The else if statement encloses an if statement within an else blocks.