

Lesson:



Fundamentals of Java



List of Concepts Involved:

- Introduction to Programming
- Basic features of Java
- Main method
- Classes and Objects (Basics)

Introduction to Programming

Making a set of instructions that instruct a computer how to carry out a task is the process of programming. There are numerous computer programming languages available for use in programming.

Category of Programming Languages:

- Machine Level Language
- Assembly language
- High-level languages

Machine Level Language: Low-level programming languages include machine language. It is also known as object code or machine code. Because machine language is typically displayed in binary or hexadecimal form (base 16) form, it is simpler to read. Since computers can read machine language programmes directly, there is no need for a translator to convert the programmes.

Machine language has the benefit of allowing programmers to execute their code more quickly than they can using high-level programming languages.

Assembly language: Another form of low-level programming language created for certain processors is assembly language (ASM). It gives the collection of directions in a symbolic and clear manner for humans to understand. An assembler is used to translate assembly language into machine language. Assembly language has the benefit of requiring less memory and less time to execute a program.

High-level languages: The high-level programming language (HLL) is made for creating user-friendly web pages and software applications. The program must be converted into machine code using a compiler or interpreter when using this programming language (execute the program). A high-level language's main benefit is that it is simple to read, write, and maintain.

Python, Java, JavaScript, PHP, C#, C++, Objective C, Cobol, Perl, Pascal, LISP, FORTRAN, and Swift are examples of high-level programming languages.

Why do we need Programming?

Programming Language is important in our daily life to enhance and increase the power of computers, mobile solutions, and the internet. There are numerous examples you may come to know when you are going to learn a programming language.

- Not just for academics, but for the real world as well, all programming is done.
- Due to programmes developed by computer programming, you can conduct online banking and purchase your ticket while travelling by train or aeroplane. It's true that your washing machine has a few different kinds of computer programmes. Programming makes it possible for all of these things and many more.

Introduction to Java

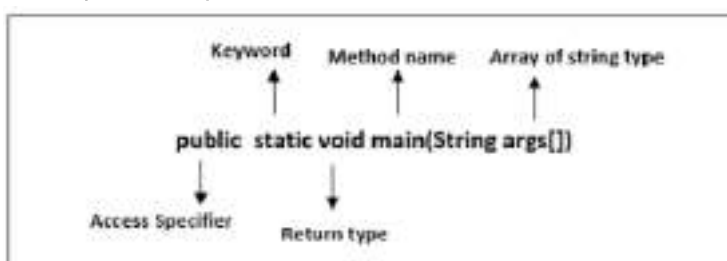
- Java is an object-oriented programming language developed by Sun Microsystems, and it was released in 1995.
- James Gosling initially developed Java in Sun Microsystems (which was later merged with Oracle Corporation).
- Java is a set of features of C and C++. It has obtained its format from C, and OOP features from C++.
- Java programs are platform independent which means they can be run on any operating system with any processor as long as the Java interpreter is available on that system.
- Java code that runs on one platform does not need to be recompiled to run on another platform; it's called write once, run anywhere(WORA).
- Java Virtual Machine (JVM) executes Java code, but it has been written in platform-specific languages such as C/C++/ASM, etc. JVM is not written in Java and hence cannot be platform independent, and Java interpreter is a part of JVM.

Features Of Java

- Object-Oriented - The features of object-oriented programming are supported by Java. Its object model is straightforward and flexible.
- Platform independent - Because Java and C++ are platform independent, application programs created in one Operating system can run on any other Operating system. C and C++, however, are platform dependent languages, making it impossible for application programs created in one Operating system to run in any other Operating system.
- Simple - Because Java incorporates many C/C++ capabilities, it is simple to understand.
- Secure - Java offers a variety of defences against malware and viruses. It guarantees that neither damage nor security will be compromised.
- Portable - We have the idea of portability in Java. Java allows the same software to run on various platforms.
- Robust - It assists us in identifying potential errors as soon as feasible during program development.
- Multi-threaded - Java's multithreading programming capability enables you to create a program that executes multiple tasks concurrently.
- Distributed - Java maintains the TCP/IP protocol and is therefore suitable for distributed Internet environments.

Main method in Java:

The main () is the starting point for JVM to start execution of a Java program. Without the main () method, JVM will not execute the program. The syntax of the main () method is: public: It is an access specifier. We should use a public keyword before the main () method so that JVM can identify the execution point of the program.



public: An access specifier, that is. Before calling the `main()` method, we need to use the `public` keyword to let the JVM know where the programme is actually being executed. Before the `main()` method, if we use `private`, `protected`, and `default`, the JVM won't be able to see it.

static: You can make a method static by using the keyword `static`. We should call the `main()` method without creating an object. Static methods are the method which is invoked without creating the objects, so we do not need any object to call the `main()` method.

void: In Java, every method has the return type. Void keyword acknowledges the compiler that the `main()` method does not return any value.

main(): It is a default signature which is predefined in the JVM. It is called by JVM to execute a program line by line and end the execution after completion of this method. We can also overload the `main()` method.

String args[]: The `main()` method also accepts some data from the user. It accepts a group of strings, which is called a string array. It is used to hold the command line arguments in the form of string values.

```
main(String args[])
```

Here, `args[]` is the array name, and it is of `String` type. It means that it can store a group of strings. Remember, this array can also store a group of numbers but in the form of string only. Values passed to the `main()` method are called arguments. These arguments are stored into an `args[]` array, so the name `args[]` is generally used for it.

Classes and Objects (Basics)

What is an Object?

An object is an entity with state and behaviour, such as a chair, bike, marker, pen, table, or car. It could be intellectual or physical (tangible and intangible). The banking system is an illustration of an intangible entity.

An object has three characteristics:

- **State:** represents an object's data (value).
- **Behavior:** represents how an object behaves (or how it functions), such as when you deposit or withdraw money.
- **Identity:** Usually, a distinct ID is used to implement an object's identification. The external user cannot see the value of the ID. However, the JVM uses it internally to uniquely identify each object.

Pen, for instance, is an object. Reynolds is its name; its state is noted as being white. Writing is its behaviour because it is utilized for writing.

A class's instances are objects. A class serves as a model or blueprint from which new objects can be made. Therefore, a class's instance (or result) is an object.

Different ways of Object Definitions

- An object is a real-world entity.
- An object is a runtime entity.
- The object is an entity which has state and behaviour.
- The object is an instance of a class.

What is a Class in Java?

A class is a collection of items with similar characteristics. It serves as a model or blueprint from which things can be made. It makes sense as a whole. It cannot be bodily.

In Java, a class could include:

- Fields
- Methods
- Constructors
- Blocks
- Nested class and interface

