# FAQ

**1. What is java?**

Java is object orient programming language.

It is simple, distributed, interpreted, robust, secure, platform independent(Architecture-Neutral), portable, high-performance, multithreaded and dynamic programming language.

In java, everything is an object.

**Simple:** Removed programming hurdles form previous languages and added garbage collection and predefined APIs.

**object orient:**

What is a class in java?

A class is a collection of data and methods that operate on that data.

The data and methods describe the state and behavior of an object.

**distributed**: Java has transparent RPC system.

Java supports various levels of network connectivity through classes in the java.net package.

**Interpreted:** Java compiler generates byte-codes. That can be execute using java interpreter(JRE) in any platform.

**Robust:** Java is portable across many Operating systems and also it has feature of Automatic memory management and garbage collection.

It's strong type checking mechanism helps in making it Robust. Especially system crashing bugs, are very rare in Java.

**Secure:** JVM is an interpreter, when the byte codes are executed it can take care of the security based on access restrictions.

**Architecture-Neutral:** The java byte code is platform independent. It has nothing do with operation system architecture while compiling source and executing byte code.

**Portable:** Java byte code is platform independent and it can run only in JVM.

**High-Performance:** Java uses Just-in-time compiler to to achieve high performance. It can compile large programs quickly.

**Multithreaded:** Java is inherently multi-threaded. A single Java program can have many different threads executing independently and continuously.

**Dynamic:** In java, dynamic memory allocation eliminate the waste of memory space and loss of data.Memory space is created during run time.

**2. What are the OOPs concepts implemented in java?**

Encapsulation

Inheritance

Polymorphism

Abstraction

**Encapsulation:** The class is fulfillment of encapsulation. Operating the private data through the method(s) of class.

Or

A mechanism of wrapping the data (variables) and methods acting on the data together as a single unit.

**Inheritance:** It is a mechanism deriving a class from another class.

A class derived from another class is called subclass

The class from which a subclass is derived is called a superclass.

**Polymorphism:** Performing multiple operations using the same context is called polymorphism.

There are two types of polymorphism in java. compile time polymorphism and runtime polymorphism.

compile time polymorphism can perform by method overloading. and runtime time polymorphism can perform by method overriding.

**Method overloading:** Method Overloading is a feature that allows a class to have more than one method having the same name, if their argument lists are different.

**Method overriding:** Overriding means to override the functionality of an existing method in any of its super classes.

**Abstraction:** Abstraction in Java is used to hide certain details and only show the essential features of the object.

**3. What is difference Between DOM SAX Parsers?**

**DOM**

Tree model parser (Object based) (Tree of nodes).

DOM loads the file into the memory and then parse- the file.

Has memory constraints since it loads the whole XML file before parsing.

DOM is read and write (can insert or delete nodes).

If the XML content is small, then prefer DOM parser.

Backward and forward search is possible for searching the tags and evaluation of the information inside the tags. So this gives the ease of navigation.

Slower at run time.

**SAX**

Event based parser (Sequence of events).

SAX parses the file as it reads it, i.e. parses node by node.

No memory constraints as it does not store the XML content in the memory.

SAX is read only i.e. can’t insert or delete the node.

Use SAX parser when memory content is large.

SAX reads the XML file from top to bottom and backward navigation is not possible.

Faster at run time.