OOP-Unit-1 Notes

What is Object-Oriented Programming?

The word **object-oriented** is the combination of two words i.e. **object** and **oriented**. The dictionary meaning of the object is an article or entity that exists in the real world. The meaning of oriented is interested in a particular kind of thing or entity. In layman's terms, it is a programming pattern that rounds around an object or entity are called **object-oriented programming.**

Object-Oriented Programming or OOP refers to languages that use objects in programming. Object-oriented programming aims to implement real-world entities like inheritance, hiding, polymorphism, etc in programming. The main aim of OOP is to bind together the data and the functions that operate on them so that no other part of the code can access this data except that function.

It is the most popular programming model among developers. It is well suited for programs that are large, complex, and actively updated or maintained. It simplifies software development and maintenance by providing major concepts such as **abstraction, inheritance, polymorphism**, and **encapsulation**. These core concepts support OOP.

Pillars of OOP

The major concepts that we have discussed above are known as **pillars of OOP.** There are **four** pillars on which OOP rests.

- Abstraction
- Encapsulation
- Inheritance
- Polymorphism

Principles of object-oriented programming

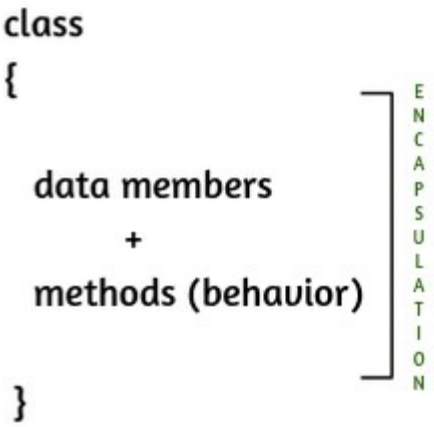
These are the four main principles of the object-oriented programming paradigm. Understanding them is essential to becoming a successful programmer.

1. Encapsulation
2. Inheritance
3. Abstraction

4. Polymorphism

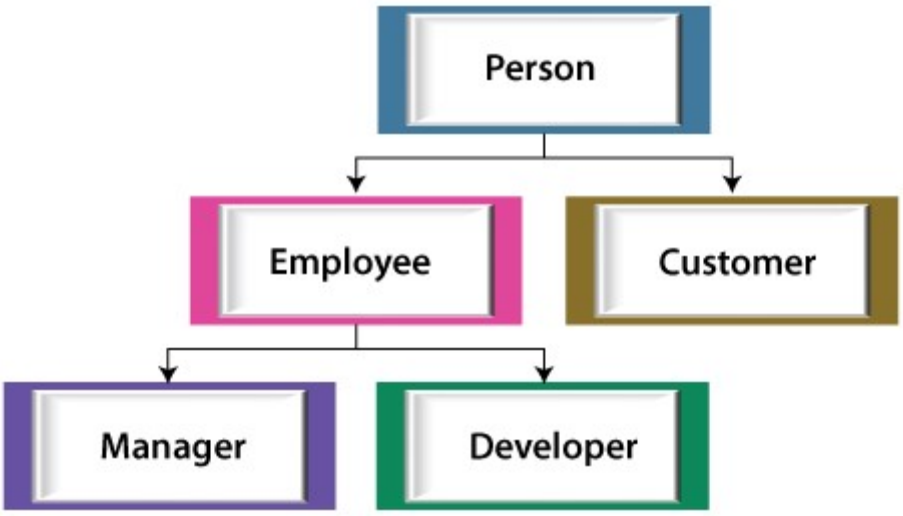
Encapsulation

Encapsulation is a mechanism that allows us to bind data and functions of a class into an entity. It protects data and functions from outside interference and misuse. Therefore, it also provides security. A class is the best example of encapsulation.



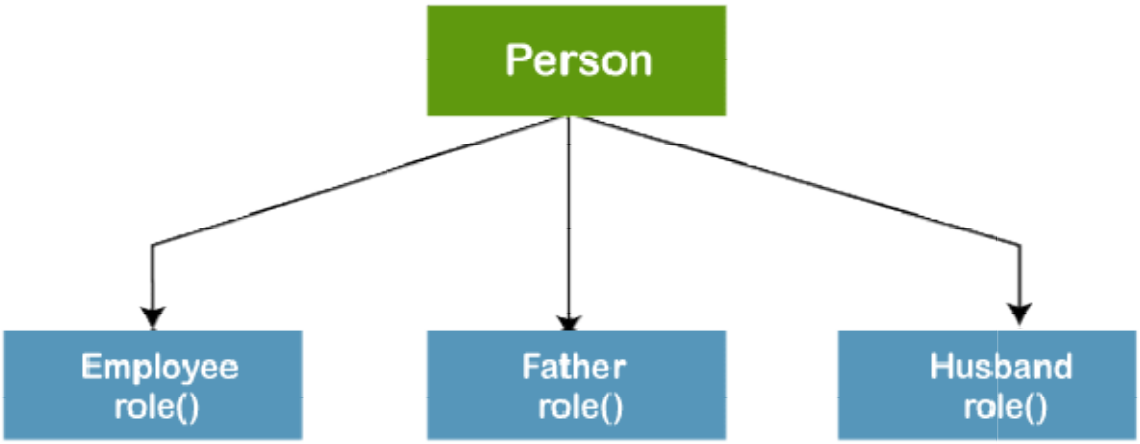
Inheritance

The concept allows us to inherit or acquire the properties of an existing class (parent class) into a newly created class (child class). It is known as **inheritance**. It provides code reusability.



Polymorphism

The word **polymorphism** is derived from the two words i.e. **ploy** and **morphs**. Poly means many and morphs means forms. It allows us to create methods with the same name but different method signatures. It allows the developer to create clean, sensible, readable, and resilient code.



The above figure best describes the concepts of polymorphism. A person plays an employee role in the office, father and husband role in the home.

What is Java?

Java is a **programming language** and a **platform**. Java is a high level, robust, object-oriented and secure programming language.

Java was developed by *Sun Microsystems* (which is now the subsidiary of Oracle) in the year 1995. *James Gosling* is known as the father of Java. Before Java, its name was *Oak*. Since Oak was already a registered company, so James Gosling and his team changed the name from Oak to Java.

Platform: Any hardware or software environment in which a program runs, is known as a platform. Since Java has a runtime environment (JRE) and API, it is called a platform.

Importance of Java

One of the essential reasons for **Java's** popularity is the cross-platform compatible and built-in security. Java program can run on any machine with a Java Runtime Environment (JRE) installed. Programs operate on various computers. Java is used by many banks, manufacturers, insurance organizations, utilities, and retailers. It is the reason that major industries ruled by Java.

In this section, we will discuss **what makes Java so popular** and what are the major industries that uses Java programming language.

Features that Makes Java Popular

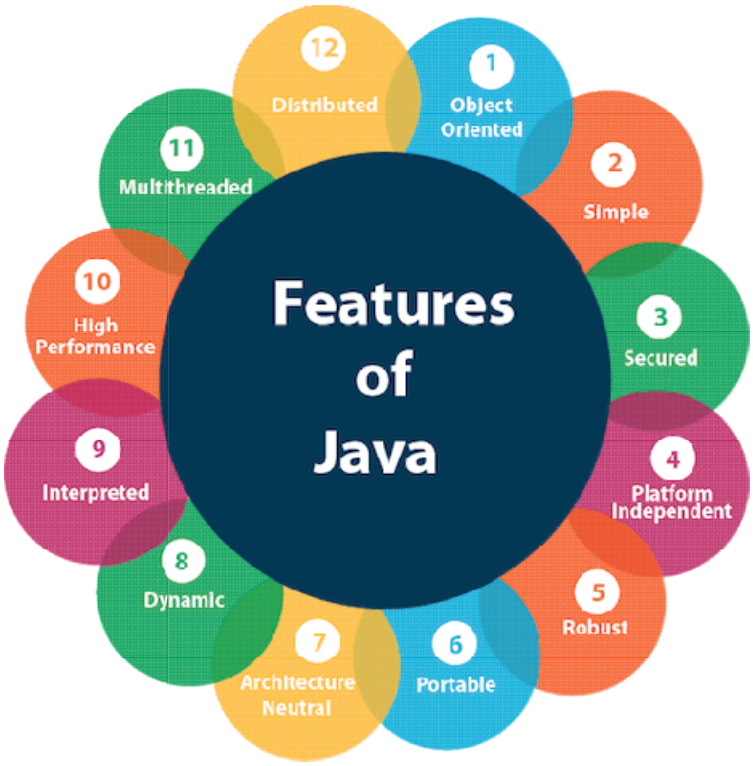
The following features make the Java programming language special and popular.

- Simple to use
- Built-in Security
- Open Source
- Robust API
- Strong community
- Excellent documentation
- Powerful set of Programming Tools
- Versatility

Features of Java

The primary objective of Java programming language creation was to make it portable, simple and secure programming language. Apart from this, there are also some excellent features which play an important role in the popularity of this language. The features of Java are also known as Java buzzwords.

A list of the most important features of the Java language is given below.



Java Industry Usage:

Java is used in almost all fields, be it financial, e-commerce, enterprise, mobile, distributed, or big data applications. For example, most of the financial software used by big players like Citigroup, Barclays, etc. is Java-based. E-commerce giant Amazon uses Java-based applications for its operations

Compilation and Running of a Java Program

1: Compilation

First, the source '.java' file is passed through the compiler, which then encodes the source code into a machine-independent encoding, known as Bytecode. The content of each class contained in the source file is stored in a separate '.class' file.

2: Execution

The class files generated by the compiler are independent of the machine or the OS, which allows them to be run on any system. To run, the main class file (the class that contains the method main) is passed to the JVM and then goes through three main stages before the final machine code is executed. These stages are:
These states do include:

1. ClassLoader

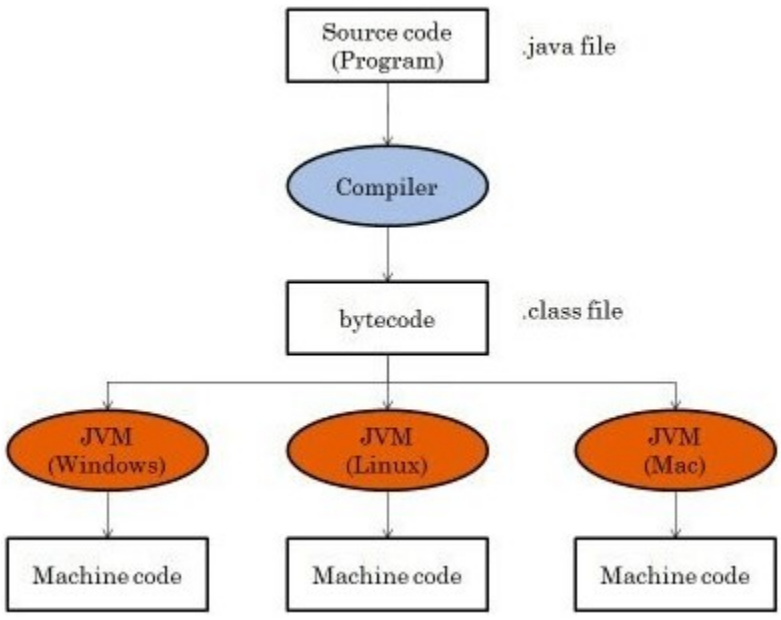
2. Bytecode Verifier
3. Just-In-Time Compiler

What is Java Bytecode?

Java bytecode is the instruction set for the Java Virtual Machine. It acts similar to an assembler which is an alias representation of a C++ code. As soon as a java program is compiled, java bytecode is generated. In more apt terms, java bytecode is the machine code in the form of a .class file. With the help of java bytecode we achieve platform independence in java.

How does it works?

When we write a program in Java, firstly, the compiler compiles that program and a bytecode is generated for that piece of code. When we wish to run this .class file on any other platform, we can do so. After the first compilation, the bytecode generated is now run by the Java Virtual Machine and not the processor in consideration. This essentially means that we only need to have basic java installation on any platforms that we want to run our code on. Resources required to run the bytecode are made available by theJava Virtual Machine, which calls the processor to allocate the required resources. JVM's are stack-based so they stack implementation to read the codes.



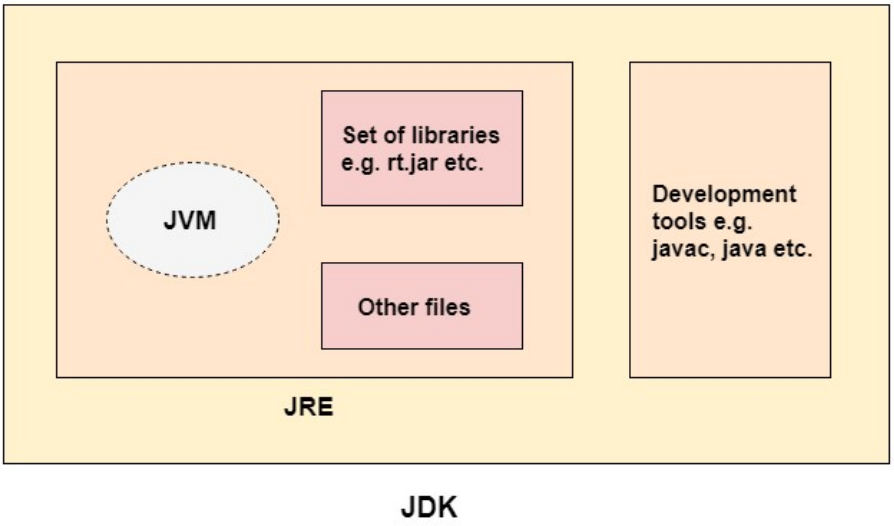
JDK

JDK is an acronym for Java Development Kit. The Java Development Kit (JDK) is a software development environment which is used to develop Java applications and applets. It physically exists. It contains JRE + development tools.

JDK is an implementation of any one of the below given Java Platforms released by Oracle Corporation:

- Standard Edition Java Platform
- Enterprise Edition Java Platform
- Micro Edition Java Platform

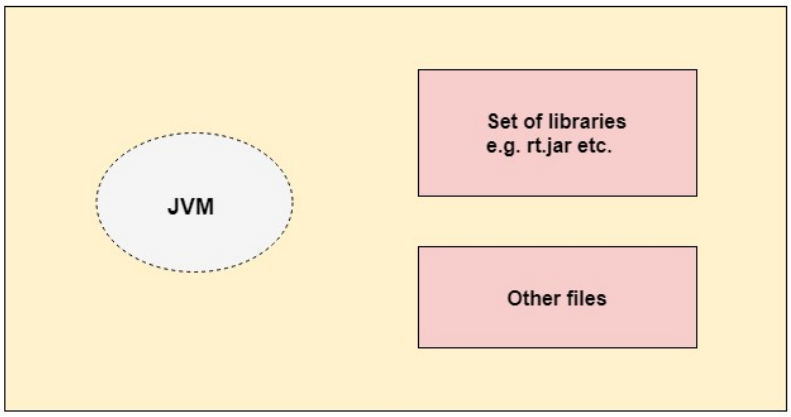
The JDK contains a private Java Virtual Machine (JVM) and a few other resources such as an interpreter/loader (java), a compiler (javac), an archiver (jar), a documentation generator (Javadoc), etc. to complete the development of a Java Application.



JRE

JRE is an acronym for Java Runtime Environment. It is also written as Java RTE. The Java Runtime Environment is a set of software tools which are used for developing Java applications. It is used to provide the runtime environment. It is the implementation of JVM. It physically exists. It contains a set of libraries + other files that JVM uses at runtime.

The implementation of JVM is also actively released by other companies besides Sun Microsystems



JRE

JVM (Java Virtual Machine)

JVM (Java Virtual Machine) is an abstract machine. It is a specification that provides runtime environment in which java bytecode can be executed.

JVMs are available for many hardware and software platforms (i.e. JVM is platform dependent).

It is:

1. **A specification** where working of Java Virtual Machine is specified. But implementation provider is independent to choose the algorithm. Its implementation has been provided by Oracle and other companies.
2. **An implementation** Its implementation is known as JRE (Java Runtime Environment).
3. **Runtime Instance** Whenever you write java command on the command prompt to run the java class, an instance of JVM is created.

What it does

The JVM performs following operation:

- Loads code
- Verifies code
- Executes code
- Provides runtime environment

JVM provides definitions for the:

- Memory area
- Class file format
- Register set
- Garbage-collected heap
- Fatal error reporting etc.