CORE JAVA

History of Java

- **James Gosling**, **Mike Sheridan**, and **Patrick Naughton** initiated the Java language project in June 1991. The small team of sun engineers called **Green Team**.
- Originally designed for small, embedded systems in electronic appliances like set-top boxes.
- Firstly, it was called **"Greentalk"** by James Gosling and file extension was .gt.
- After that, it was called **Oak** and was developed as a part of the Green project.

What is Java?

- Java is a **programming language** and a **platform**.
- Java is a high level, robust, secured and object-oriented programming language.
- **Platform**: Any hardware or software environment in which a program runs, is known as a platform. Since Java has its own runtime environment (JRE) and API, it is called platform.

Features of Java

There is given many features of java. They are also known as java buzzwords. The Java Features given below are simple and easy to understand.

1. Simple
2. Object-Oriented
3. Portable
4. Platform independent
5. Secured
6. Robust
7. Architecture neutral
8. Dynamic
9. Interpreted
10. High Performance
11. Multithreaded
12. Distributed

- Simple:
  - According to Sun, Java language is simple because:
  - syntax is based on C++ (so easier for programmers to learn it after C++).
  - removed many confusing and/or rarely-used features e.g., explicit pointers, operator overloading etc.
  - No need to remove unreferenced objects because there is Automatic Garbage Collection in java.

- Object-oriented:
  - Object-oriented means we organize our software as a combination of different types of objects that incorporates both data and behaviour.
  - Object-oriented programming(OOPs) is a methodology that simplify software development and maintenance by providing some rules.
  - Basic concepts of OOPs are:

1. Object
2. Class
3. Inheritance
4. Polymorphism
5. Abstraction
6. Encapsulation

- Platform Independent:
  - A platform is the hardware or software environment in which a program runs.
  - There are two types of platforms software-based and hardware-based. Java provides software-based platform.
  - The Java platform differs from most other platforms in the sense that it is a software-based platform that runs on the top of other hardware-based platforms. It has two components:

    - ✓ Runtime Environment
    - ✓ API(Application Programming Interface)

- ▪ Secured:

  - ➢ Java is secured because:

    - ✓ **No explicit pointer**
    - ✓ **Java Programs run inside virtual machine sandbox**
    - ✓ **Classloader:** adds security by separating the package for the classes of the local file system from those that are imported from network sources.
    - ✓ **Bytecode Verifier:** checks the code fragments for illegal code that can violate access right to objects.
    - ✓ **Security Manager:** determines what resources a class can access such as reading and writing to the local disk.

- ▪ Robust:

  - ➢ Robust simply means strong. Java uses strong memory management. There are lack of pointers that avoids security problem. There is automatic garbage collection in java. There is exception handling and type checking mechanism in java. All these points makes java robust.

- ▪ Architecture-neutral:

  - ➢ There is no implementation dependent features e.g. size of primitive types is fixed.
  - ➢ In C programming, int data type occupies 2 bytes of memory for 32-bit architecture and 4 bytes of memory for 64-bit architecture. But in java, it occupies 4 bytes of memory for both 32 and 64 bit architectures.

- ▪ Portable:

  - ➢ We may carry the java bytecode to any platform.

- High-performance:
    - Java is faster than traditional interpretation since byte code is "close" to native code still somewhat slower than a compiled language (e.g., C++)

- Distributed:
    - We can create distributed applications in java. RMI and EJB are used for creating distributed applications. We may access files by calling the methods from any machine on the internet.

- Multi-threaded:
    - A thread is like a separate program, executing concurrently. We can write Java programs that deal with many tasks at once by defining multiple threads. The main advantage of multi-threading is that it doesn't occupy memory for each thread. It shares a common memory area. Threads are important for multi-media, Web applications etc.

Java Example

Let's have a quick look at java programming example. A detailed description of hello Java example is given in next page.

```
class Simple{
   public static void main(String args[]){
    System.out.println("Hello Java");
   }
}
```

What are the uses of Java?

According to Sun, 3 billion devices run java. There are many devices where Java is currently used. Some of them are as follows:

1. Desktop Applications such as acrobat reader, media player, antivirus etc.
2. Web Applications such as irctc.co.in, javatpoint.com etc.
3. Enterprise Applications such as banking applications.
4. Mobile
5. Embedded System
6. Smart Card
7. Robotics
8. Games etc.

Which Types of Applications can be developed by Java?

There are mainly 4 types of applications that can be created using java programming:

- Standalone Application
    - ➢ It is also known as desktop application or window-based application. An application that we need to install on every machine such as media player, antivirus etc. AWT and Swing are used in java for creating standalone applications.
- Web Application
    - ➢ An application that runs on the server side and creates dynamic page, is called web application. Currently, servlet, jsp, struts, jsf etc. technologies are used for creating web applications in java.
- Enterprise Application
    - ➢ An application that is distributed in nature, such as banking applications etc. It has the advantage of high level security, load balancing and clustering. In java, EJB is used for creating enterprise applications.

- Mobile Application

    - An application that is created for mobile devices. Currently Android and Java ME are used for creating mobile applications.

Java Latest Version History

There are many java versions that has been released. Current stable release of Java is Java SE 8.

1. JDK Alpha and Beta (1995)
2. JDK 1.0 (23rd Jan, 1996)
3. JDK 1.1 (19th Feb, 1997)
4. J2SE 1.2 (8th Dec, 1998)
5. J2SE 1.3 (8th May, 2000)
6. J2SE 1.4 (6th Feb, 2002)
7. J2SE 5.0 (30th Sep, 2004)
8. Java SE 6 (11th Dec, 2006)
9. Java SE 7 (28th July, 2011)
10. Java SE 8 (18th March, 2014)