FACULTY OF Engineering & Technology

What is Java

1. Java is a High level **programming language** and a **platform independent language**.
2. Java is a robust, object-oriented and secure programming language.

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

Types of Java Applications

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

1) **Standalone Application**

Standalone applications are also known as desktop applications or window-based applications. These are traditional software that we need to install on every machine. Examples of standalone application are Media player, antivirus, etc. AWT and Swing are used in Java for creating standalone applications.

2**) Web Application**

An application that runs on the server side, creates a dynamic pageand provide services over web is called a web application. Servlet, jsp etc are web technologies used for creating web applications in Java.

3**) Enterprise Application**

An application that is distributed in nature, such as banking applications, etc. is called enterprise application. It has advantages of the high-level security, load balancing, and clustering. In Java, EJB is used for creating enterprise applications.

4**) Mobile Application**

An application which is created for mobile devices is called a mobile application. Currently, Android and Java ME are used for creating mobile applications

Java Editions:

1. J2SE or JSE
2. J2EE or JEE
3. J2ME or JME

History of JAVA

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

Features of Java

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