Lec 5 History

Java became popular because it allowed programs to run on any operating system, thanks to its platform independence. This was especially important after the internet connected the world, making it possible to share and use software globally. To start programming in Java today, you need to install the Java Development Kit (JDK) and a code editor like Eclipse.

Java History and Platforms

Java's development included multiple platforms, such as Java Standard Edition (SE) and Java Enterprise Edition (EE, formerly J2EE). The concept of "core Java" can vary, but it generally refers to Java SE, which provides the basic language features.

The Origins of Java and Python

Python was created in 1991, while Java's first version appeared in 1996. Java was developed by Sun Microsystems, a large company, whereas Python was not backed by a major corporation at its start.

Java’s Early Popularity

Java gained popularity quickly due to strong backing from Sun Microsystems, which provided marketing, infrastructure, and a dedicated team, unlike languages developed by individuals or smaller groups.

Pre-Java Programming Landscape

Before Java, C and C++ were the most popular languages, but they were platform dependent, meaning programs ran only on specific operating systems. This was not a big problem before global connectivity.

Limited Global Connectivity Before 1990

Before the internet and web browsers, countries and companies worked in isolation. Software was sold and used locally, and there was little need for cross-platform compatibility.

The Birth of the Web and Global Change

The first web browser and the World Wide Web were created around 1990-1991, connecting the world and enabling global software sharing and communication.

Ecosystems and Platform Dependence

Companies and users within the same region adopted the same operating systems and programming languages, making sharing and using software easy within that ecosystem. There was little incentive to switch to platform-independent languages.

The Shift: Internet and the Need for Platform Independence

With the internet connecting the world, the need arose for software that could run on any system, regardless of the operating system. Java offered this through platform independence, allowing programs to be shared and used globally.

Java’s Rise with the World Wide Web

Java became popular because its platform independence matched perfectly with the global reach of the World Wide Web. Developers could write code once and have it run anywhere, making Java the preferred choice for internet-era applications.

Transition to Technical Concepts

After covering history and popularity, the focus will shift to technical topics like how Java programs run,

what bytecode and JVM are, and the flow of program execution.

Installing Java Development Kit (JDK)

To start programming in Java, you must install the Java Development Kit (JDK), which includes the compiler and libraries needed for development. Java 17 is recommended due to its compatibility with older code and modern frameworks.

Choosing a Code Editor or IDE

The JDK does not come with a code editor, so you need a separate tool to write and manage code. Notepad can be used, but Integrated Development Environments (IDEs) like Eclipse, IntelliJ, or VS Code make coding much easier with helpful features.

Downloading and Setting Up Eclipse

Eclipse is a popular free IDE for Java. Download the latest version of Eclipse for your operating system and extract the zip file. No installation is needed—just extract and run the application.

Testing Java Installation

After installing Java, open the command prompt and type java -version to confirm the installation. Eclipse should only be opened after Java is set up to avoid errors.

Preparing for Your First Java Program

The next step will be to write a simple Java program first using Notepad and then using Eclipse, to understand the benefits of an IDE and the basics of Java program execution.