COMMENTING ON JAVA 8 FEATURES

TYPES OF JAVA-8 FEATURES:

Java 8 introduced many new features and enhancements, some of the major ones are:

1. Lambda expressions: A new syntax that allows the creation of anonymous functions in Java.
2. Stream API: A new set of classes and interfaces that provide a functional programming-style approach to processing collections of data.
3. Default methods: Allows interfaces to provide implementation for their methods.
4. Functional interfaces: Interfaces that have a single abstract method and can be used with lambda expressions.
5. Optional class: A new class that provides a way to represent values that may or may not be present.
6. Date and Time API: A new set of classes and interfaces for working with dates and times.
7. Method references: A shorthand syntax for lambda expressions that refer to existing methods.
8. Type annotations: A new way to specify annotations on types.
9. Nashorn JavaScript engine: A new lightweight JavaScript engine that can be used to execute JavaScript code from within Java.
10. PermGen space removed: The permanent generation (PermGen) space was removed, and the metadata is now stored in the native memory.
11. Concurrency updates: Updates to the java.util.concurrent package, including new classes and methods for working with concurrent programming.
12. Compact profiles: A new feature that allows Java developers to create smaller Java runtime environments by including only the necessary classes and packages.

These are some of the major features that were introduced in Java 8.

IMPORTANCE OF JAVA 8 :

1. **It enables the Internet of Things**. Java 8 can help businesses take part in the world of connected, always-on devices. Think smart appliances in the home and billions of wearable devices, medical sensors, connected vehicles, smart meters, and industrial controllers. The opportunities are endless. Both Java SE 8 and Java ME 8—the two platforms that make up

Java 8, along with Oracle 's related embedded products—provide a scalable, flexible, secure development platform for the Internet of Things.

2. **Less code means more productivity**. Java 8 facilitates improved developer productivity through reduced boilerplate code. This is due in large part to lambda expressions, which are the most significant and anticipated new feature in Java SE 8. Lambdas (for short) let developers simplify the code they write every day. Developers can expect the way they do their jobs to change: They will be able to write code that is more compact, simpler, and easier to maintain.
3. **Modernize your apps**. Java 8 takes a giant step forward in modernizing the Java language and Java libraries. One example is a new date and time application programming interface that reduces the complexity for developers when handling date and time, especially when dealing with internationalization and localization for different markets. The existing date and time library was more than 15 years old. It has been re-engineered using modern design practices.
4. **Embedded technology is a big deal.** Developers building embedded apps can use one of three new Compact Profiles in Java SE 8. These predefined subsets of the full Java SE 8 specification support the creation of smaller applications for resource-constrained devices. That means Java SE can be deployed on embedded devices using as little as a 10 MB static footprint and 16 MB of RAM.
5. **Create eye-popping graphics.** JavaFX 8 is a graphics toolkit included in JDK 8 that lets developers build rich client applications using standard Java development tools. It includes an embedded graphics stack, new UI

controls and 3D graphics features, and HTML5 support. JavaFX 8 also brings a fresh, modern look to applications with its new Modena theme.

6. **Java 8 is integrated with JavaScript.** Java and JavaScript are not the same thing. JavaScript is a popular language with features similar to Java's that is ideal for transmitting code across networks. Java SE 8 includes Nashorn, a JavaScript engine that runs on the Java Virtual Machine (JVM) and allows Java applications to contain components written in JavaScript. When developers want to use both Java and JavaScript, Nashorn can deliver significant performance improvements and interoperability between Java and JavaScript code.
7. **A worldwide community is behind Java 8.** The Java 8 release represents a huge collaborative effort by the worldwide Java community. Java user groups provided ongoing feedback to Specification Leads, made requests for functionality, uncovered bugs, and contributed to code fixes. The OpenJDK community provided a way for developers to **collaborate** on the open source reference implementation of the Java SE platform. And the Java Community Process—with representation from **the** likes of IBM , Intel , Red Hat , and SAP —provided a means for organizations and individuals to contribute to the Java platform.

INSTALLATION GUIDE FOR JDK-8 IN WINDOWS:

"JDK Installation for Microsoft Windows"

FOR OS X:

"JDK 8 Installation for OS X"