#### **Fundamentals of Java Quiz**

7 out of 8 correct

1.

1. Who invented Java Programming?

* Guido van Rossum
* James Gosling
* Dennis Ritchie
* Bjarne Stroustrup

Explanation: Java programming was developed by James Gosling at Sun Microsystems in 1995. James Gosling is well known as the father of Java.

2.

Which statement is true about Java?

* Java is a sequence-dependent programming language
* Java is a code dependent programming language
* Java is a platform-dependent programming language
* Java is a platform-independent programming language

Explanation: Java is called ‘Platform Independent Language’ as it primarily works on the principle of ‘compile once, run everywhere’.

3.

Which component is used to compile, debug and execute the java programs?

* JRE
* JIT
* JDK
* JVM

NOTE: [**Difference between JDK, JRE, and JVM**](https://www.digitalocean.com/community/tutorials/difference-jdk-vs-jre-vs-jvm#difference-between-jdk-jre-and-jvm)

JDK, JRE, and JVM are core concepts of Java programming language. We don’t use these concepts in programming. But, as a Java developer, we should know about them.

### [**1. JDK**](https://www.digitalocean.com/community/tutorials/difference-jdk-vs-jre-vs-jvm#1-jdk)

Java Development Kit aka JDK is the core component of Java Environment and provides all the tools, executables, and binaries required to compile, debug, and execute a Java Program. JDK is a platform-specific software and that’s why we have separate installers for Windows, Mac, and Unix systems. We can say that JDK is the superset of JRE since it contains JRE with Java compiler, debugger, and core classes.

### [**2. JVM**](https://www.digitalocean.com/community/tutorials/difference-jdk-vs-jre-vs-jvm#2-jvm)

JVM is the heart of Java programming language. When we execute a Java program, JVM is responsible for converting the byte code to the machine-specific code. JVM is also platform-dependent and provides core java functions such as memory management, garbage collection, security, etc. JVM is customizable and we can use java options to customize it. For example, allocating minimum and maximum memory to JVM. JVM is called virtual because it provides an interface that does not depend on the underlying operating system and machine hardware. This independence from hardware and the operating system makes java program write-once-run-anywhere.

### [**3. JRE**](https://www.digitalocean.com/community/tutorials/difference-jdk-vs-jre-vs-jvm#3-jre)

JRE is the implementation of JVM. It provides a platform to execute java programs. JRE consists of JVM, Java binaries, and other classes to execute any program successfully. JRE doesn’t contain any development tools such as Java compiler, debugger, JShell, etc. If you just want to execute a java program, you can install only JRE. You don’t need JDK because there is no development or compilation of java source code is required. Now that we have a basic understanding of JDK, JVM, and JRE, let’s look into the difference between them.

Explanation: JDK is a core component of Java Environment and provides all the tools, executables and binaries required to compile, debug and execute a Java Program.

4.

Which one of the following is not a Java feature?

* Object-oriented
* Use of pointers
* Portable
* Dynamic and Extensible

Explanation: Pointers is not a Java feature. Java provides an efficient abstraction layer for developing without using a pointer in Java. Features of Java Programming are Portable, Architectural Neutral, Object-Oriented, Robust, Secure, Dynamic and Extensible, etc.

5.

Which environment variable is used to set the java path?

* MAVEN\_Path
* JavaPATH
* JAVA
* JAVA\_HOME

Explanation: JAVA\_HOME is used to store a path to the java installation.

6.

Which of the following options leads to the portability and security of Java?

* Bytecode is executed by JVM
* The applet makes the Java code secure and portable
* Use of exception handling
* Dynamic binding between objects

Explanation: The output of the Java compiler is bytecode, which leads to the security and portability of the Java code. It is a highly developed set of instructions that are designed to be executed by the Java runtime system known as Java Virtual Machine (JVM).

7.

What is an object?

* The object is an instance of a class
* An object is a real-world entity.
* An object is a runtime entity.
* All of the above

EXPLAINATION : <https://docs.oracle.com/javase/tutorial/java/concepts/object.html>

8.

public is an access specifier?

* Yes
* No
* Name
* None of the above

Explanation: public is an access specifier, that is. Before calling the main() method, we need to use the public keyword to let the JVM know where the programme is actually being executed. Before the main() method, if we use private, protected, and default, the JVM won't be able to see it.

Submit