JAVA PROGRAMMING

ASSIGNMENT – 1

JAVA LANGUAGE :

1. Java is a platform-independent programming language. It means that we can run Java on the platforms that have a **Java interpreter**.
2. It is the reason that makes the Java platform-independent. The Java interpreter converts the Java bytecode (.class file) into the code understand by the operating system.
3. In this section, we will understand **what is an interpreter in Java, the features of the interpreter,** and **how does the Java interpreter work.**



4. We will also see **how it is different from a compiler.**



DIFFERENCE BETWEEN INTERPRETER AND COMPILER :

1. Interpreter :

1. **Java interpreter** is a computer program (system software) that implements the JVM. It is responsible for reading and executing the program.
2. It is designed in such a way that it can read the source program and translate the source code instruction by instruction.
3. **It converts the high-level program into assembly language** (machine language).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| To convert the byte code into machine code, we deploy the .class file on the [Java Virtual Machine (JVM)](https://www.javatpoint.com/jvm-java-virtual-machine). The JVM converts that code into machine code using the Java interpreter. The JVM uses the interpreter at runtime, after that it execute the code on the host machine.  4.9M  78  Prime Ministers of India | List of Prime Minister of India (1947-2020)  Java Interpreter  As the Java compiler compiles the source code into the [Java bytecode](https://www.javatpoint.com/java-bytecode). In the same way, the Java interpreter converts or translates the bytecode into the machine-understandable format i.e. machine code, after that the machine code interacts with the operating system.  If the JVM is installed on any system it means that the platform is JVM enabled. The platform performs all the tasks of the [Java](https://www.javatpoint.com/java-tutorial) run-time system. It loads the Java class file and interprets the compiled byte-code.  The [browsers](https://www.javatpoint.com/browsers), like [Google Chrome](https://www.javatpoint.com/google-chrome), Netscape, etc. are the popular example that contains the Java interpreter. It means these are Java-enabled browsers. It is used to run the Applet in the browser. The interpreter also serves as a specialized compiler in an implementation that supports dynamic or **just-in-time (JIT)** compilation which turns the Java bytecode into native machine instructions.  Let's see how an interpreter loads a Java program.  First, we specify the class by using the **java** command followed by the class name and options available for the interpreter, and command-line arguments if required. We use the following command to load the class:   1. % java [interpreter options] **class** name [arguments]   In the above command, the class name should be a fully qualified name (the name of the class that includes the package name, if any). Remember that, we do not write the .class extension at the end of the class name. For example:   1. java Product 2. java c ulix.Product   In the first command, **Product** is the class name. In the second command, Ulix is the name of the package in which the Package class is stored.  Once the class is loaded, Java follows a convention and searches for the class that contains the main() method. When the JVM founds the main() method, the interpreter starts the application by invoking the main() method. After executing the main() method, additional threads, and references other classes.  Features of Interpreter :  It converts the source code into machine language, line by line at run time, without changing the sequence.   * An interpreter does not generate an intermediate machine code * Each error of every line is displayed one by one * When compared to a compiler, the program execution speed is slower   \* Less amount of time is spent on analyzing and processing the program   1. Compiler :      * 1. A Java compiler is a program that takes the text file work of a developer and [compiles](https://whatis.techtarget.com/definition/compiler) it into a platform-independent [Java](https://www.theserverside.com/definition/Java) file.   2. Java compilers include the Java Programming Language Compiler (javac), the GNU Compiler for Java (GCJ), the [Eclipse](https://searchapparchitecture.techtarget.com/definition/Eclipse-Eclipse-Foundation) Compiler for Java (ECJ) and  [Jikes](https://www.theserverside.com/definition/Jikes).  * 1. Programmers typically write language statements in a given programming language one line at a time using a code editor or an integrated development environment (IDE).   2. The resulting file contains what are called the source statements. The programmer then runs a compiler for the appropriate language, specifying the name of the file that contains the source statements.   3. At run time, the compiler first parses (analyzes) all of the language statements syntactically and then, in one or more successive stages or "passes,” builds the output code, making sure that statements that refer to other statements are referred to correctly in the final code.   4. Generally, Java compilers are run and pointed to a programmer’s code in a text file to produce a [class](https://whatis.techtarget.com/definition/class) file for use by the Java virtual machine ([JVM](https://www.theserverside.com/definition/Java-virtual-machine-JVM)) on different platforms. Jikes, for example, is an open source compiler that works in this way.   5. A just-in-time ([JIT](https://www.theserverside.com/definition/just-in-time-compiler-JIT)) compiler comes along with the Java VM. Its use is optional, and it is run on the platform-independent code. The JIT compiler then translates the code into the machine code for different hardware so that it is optimized for different architectures. Once the code has been (re-)compiled by the JIT compiler, it will usually run more quickly than the Java code that can only be executed one instruction at a time.   Main functions of a Java compiler :   * + 1. The compiler **translates your source code instructions into Java bytecode instructions**.     2. In other words, the compiler takes code that you can write and understand and translates it into code that a computer can execute (like the code here).     3. You might put your source code in a file named Hotel. java.   Difference Between Interpreter and Compiler :  In the following table, we have summarized the key differences between an interpreter and a compiler.   |  |  | | --- | --- | | **Interpreter** | **Compiler** | | It translates the code instruction by instruction. | It translates the entire program at once. | | Its execution is slower. | Its execution is faster. | | Its compile time is less. | It takes more time to compile the code. | | It does not generate the intermediate object code. | It generates the intermediate object code. | | It compiles the program until an error is found. | All the errors show once at the end of the compilation. | | Python, PHP, Ruby, and Perl use an interpreter. | Java, C++, Scala, and C uses a compiler. | |

