**Angeline Karen – 2440035601**

Java Compilation Process

Java compilation process doesn’t work on a one-step compilation. It doesn’t directly compile into an executable file. In Java, it goes through two-step execution: compilation and execution.

1. Compilation

Java source code is saved with the extension of ‘.java’ and it will pass through the Java compiler (Javac). Javac is a Java compiler that compiles Java code into a Java bytecode. Then, those Bytecode gets saved on the disk with the file extension ‘.class’. Java is an Object-Oriented Programming (OOP) language, therefore a program in Java is made of one or more classes.

1. Execution

During the runtime, these .class file is passed to Java Virtual Machine (JVM) to be loaded through the class loader. These classes won’t be loaded into memory all at once, but dynamically. A primordial class loader is embedded into all the JVMs and is the default class loader. After this bytecode of a class file is loaded, it goes through an inspection that is done by the bytecode verifier. It checks through to make sure there wouldn’t any instructions performing damage actions. However, if the checks failed, the verifier won’t allow the class to be loaded. Finally, the Just-In-Time compiler (JIT) will read those loaded bytecodes in sections and compiles them dynamically (same sequence repeatedly) into machine-level code. This leads the program to run/execute faster and can be cached and reused later without needing to be re-compiled.