Week1: Java Technologies

Student Activity(Graded)

You are required to write a brief history of Java technologies highlighting the major events and Java versions to date.

Your write-up should not exceed 600wordsandthefilesize1MB.

NB: Avoid plagiarizing at all cost (Make sure the write-up is your own work)

Due Date: 21st February, 2024

Introduction

In this unit we will cover Java as a technology. We will discuss Java platform, Java versions, Java translation process and Java Development Environments. A brief history of Java Technologies has been given to students as an assignment. Therefore, we will skip the history in this unit.

By the end of our discussion, students should be able to:

- Describe Java as a language as well as a platform
- Explain the major events in the history of Java (Part of assignment given)
- Describe the translation process of Java program
- Describe the characteristics of Java
- Identify the Java development tools

Java

Java is a powerful, general-purpose programming environment. It is one of the most widely used programming languages in the world, and has been exceptionally successful in business and enterprise computing.

The following are two motivations behind Java creation:

1) The need for platform-independent (architectural-neutral) language. There was a need of language that could be used to create software to be embedded in various consumer electronic devices with different CPU architectures
2) Emerging of the World Wide Web - while the desire for an architecture-neutral programming language provided the initial spark, the Internet ultimately led to Java's large-scale success. The web demanded the portable programs. The same problem that Java was initially designed to solve on a small scale could also be applied to the Internet on a large scale. This realization caused the focus of Java to switch from consumer electronics to Internet programming.

Java Versions and Editions

First version of Java was released in 1995, Java 1.02. The current version is Java 19 released in September 2022 and new version (Java 20) is expected in March, 2023. Early versions (versions 1.2 to 1.4) of Java were being referred as Java 2 (second generation), hence the version name was, for example, J2SE 1.2, J2SE 1.3, etc. We will cover what SE is soon. After version1.4, the naming of Java versions changed to Java 5, Java 6, etc.

Java comes in different editions: SE, EE, ME, JavaFX. In this module we will use Java 17 SE. Please note that you can use any Standard Edition from Java 5.

Student Activity (Ungraded but may be examined)

A student is required to describe the following Java Editions: SE, EE, ME, JavaFX

JAVA PROGRAM–SOURCECODE

Java program is written using Java language. The program in the human-readable form is called source code. The Java program source code is saved in a file with **.java** extension. We will learn how to write the Java program in the next unit.

Java Compiler

The source code is translated into a machine-like code using javac, a Java compiler. This machine-like-code is called byte code. The bytecode cannot be understood (executed) by the machine, hence requires further translation. The byte code is stored in a file with **.class** extension.

JVM and Interpreter

Java Virtual Machine (JVM) is a hypothetical computer that run Java program in bytecode form. JVM has an interpreter called **java**, which executes the byte code file (.class file). Before the program is run, the **bytecode verifier** examines their bytecodes to ensure that they're valid and do not violate Java's security restrictions. Java enforces strong security to make sure that Java programs arriving over the network do not damage your files or your system. The JVM makes the Java program more secure as it separates the program from the operating system as well as insulating it from particular hardware. This two-stage translation of Java program makes the program platform-independent and portable. That is, the class file generated on Linux machine can be run on any platform such as Windows, Mac, etc, provided the platform has JVM. Hence you **write once**, and **run anywhere** (**WORA**).

Just-In-Time Compiler (JIT)

We have seen that the JVM has interpreter, **java**, which is used to run the class file. To improve performance, the modern JVMs have also just-in-time compiler called Java HotSpot compiler, which compiles the parts of bytecodes that execute frequently into underlying computer's machine code. When the JVM encounters these compiled parts again, the faster machine-language code executes (Deitel,2015).

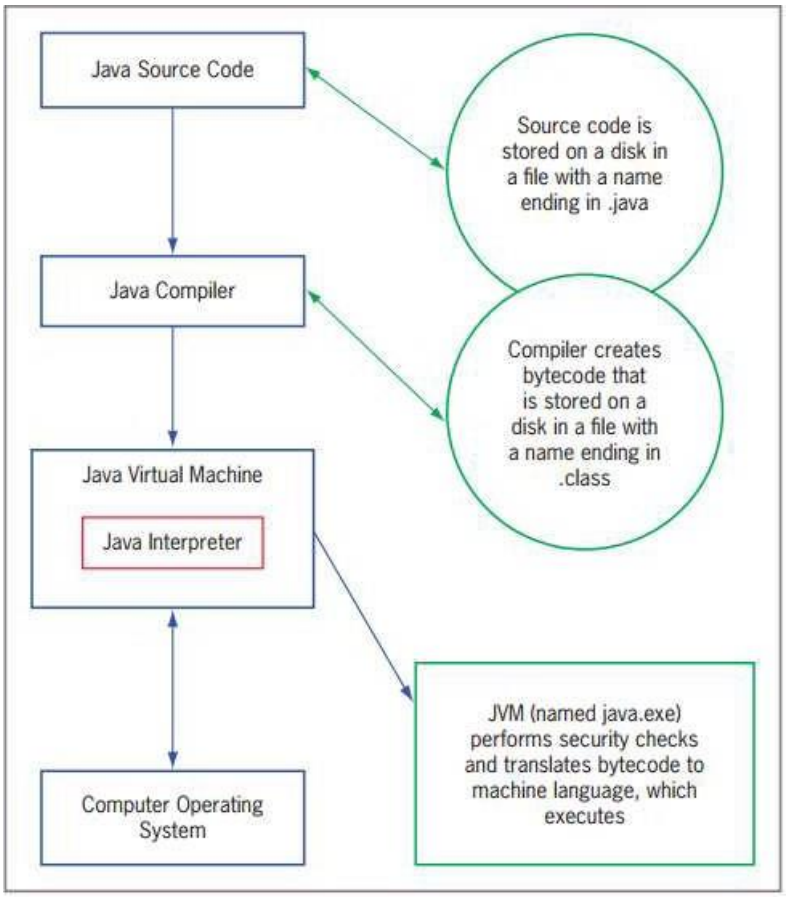


Figure 1: Java Environment (Source: Farrel, 2019,pg.11)

Characteristics of Java

The following are characteristics of Java.

- Java is Simple
- Java is Object-Oriented
- Java is Distributed
- Java is Interpreted
- Java is Robust
- Java is Secure
- Java is Architecture-Neutral
- Java is Portable
- Java's high Performance
- Java is Multithreaded
- Java is Dynamic

Student Activity (Ungraded but can be examined)

You are required to give a description for each characteristic of Java listed above.

Java Development Tools and Environments

In this section we will explore and install the Java platform and IDEs.

LAB1: Installing Java Platform

In this lab, we will install and test the Java platform (Java Development Kit, JDK).

Prerequisites: Laptop or PC with Windows 10 preferably (Windows 8 or 7 can still do)

Step1: Downloading JDK (Java Platform)

a) Go to the https://www.oracle.com/java
b) Click download
c) Scroll to Java SE17
d) Under Oracle JDK, click **JDK Download**

Step2:InstallingJavaPlatform

a) Locate the download from Step1
b) Double click it to start installation
c) Follow the wizard to finish the installation

Step3: Test that Java is added to environment variables

a) Open the terminal (Command Prompt)
b) Enter the following command

java--version

the outcome similar to the one below appears

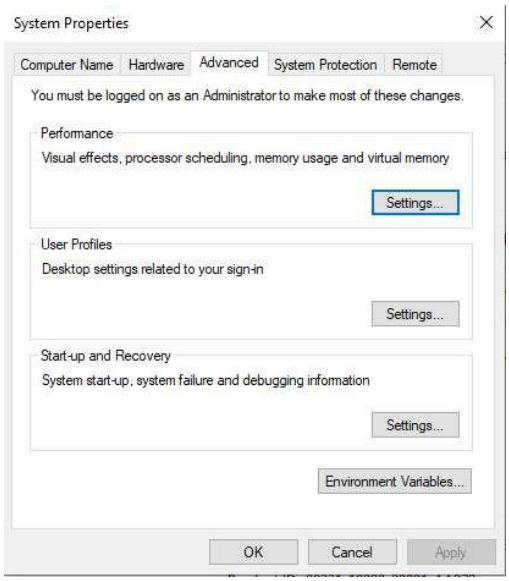
```
C:\Users\HP>java --version
java 17.0.1 2021-10-19 LTS
Java(TM) SE Runtime Environment (build 17.0.1+12-LTS-39)
Java HotSpot(TM) 64-Bit Server VM (build 17.0.1+12-LTS-39, mixed mode, sharing)
```

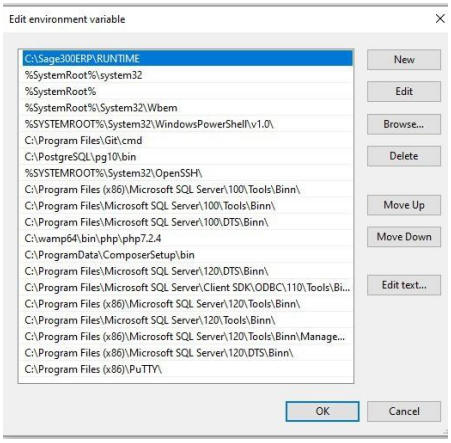
If the Command Prompt shows **'java' is not recognized as an internal or external command**, then proceed to **Step4**, otherwise proceed to **Step5**.

Step4: Add Java bin folder to the environment variable

a) Locate where Java is installed. For example, **C:\ProgramFiles\Java\jdk-17.0.2**
b) Copy the path to bin folder, **C:\Program Files\Java\jdk-17.0.2\bin**
c) Write click on **This PC** and choose **Properties**. The system window appears like the one below

d) Click on Advanced system settings on the left side of the window. The System Properties window appears.





e) Click on Environment Variables…
f) Click on **path** variable under System variables
g) Click on **Edit…**Edit environment variable window appears
h) Click **New**
i) Paste the bin location copied in b) above and click **OK**
j) Redo **Step 3: Test that Java is added to environment variables** to verify that Java path has been added successfully.

Step5: Write Welcome program

a) Open notepad or any text editor of your choice
b) Copy the program below to your text editor:

```
public class Welcome {
 public static void main(String[] args) {
 // TODO code application logic here
System.out.println("Welcome to java programming.");
                    }
            }
```

c) Create a folder and name it **java-examples** in **C:**
d) Save the **Welcome.java** to **C:\java-examples**

Step6: Compile and run Welcome program

a) Open the Command Prompt
b) Navigate to **java-examples** folder using a command below

cd C:\java-examples

c) Compile Welcome.java using a command below

Javac Welcome.java

If Command Prompt shows nothing, then you have successfully translated the program to bytecode (.**class file**). If you check your folder, you will now find another file called **Welcome.class**

d) Run the program using the command below

Java Welcome

Great!!!! You have successfully run your first Java program.

LAB2: Using Online Development Tool: REPL.IT

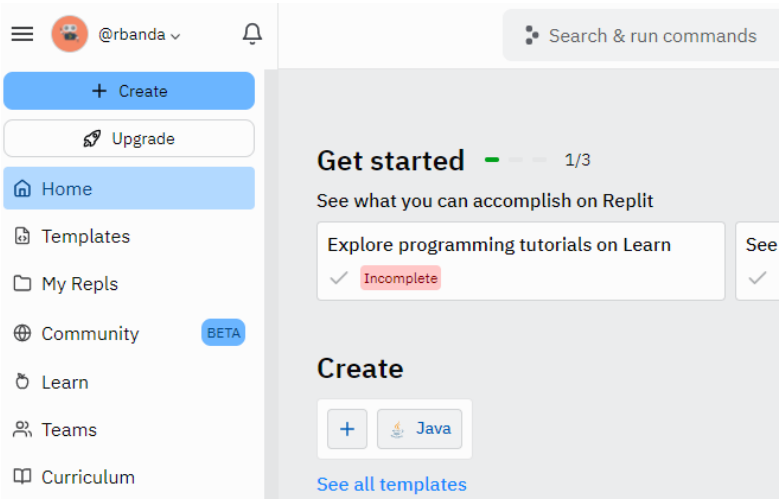
In this lab, we will create a **Repl.it** account and use it to create and run Java programs. **Repl.it** is an online program development platform (IDEs) for different programming languages. You can use it to build, host and ship applications. For Java, repl.it uses OpenJDK.

Prerequisites: Any gadget with internet access via browser such as Laptop, Desktop, and Smartphone.

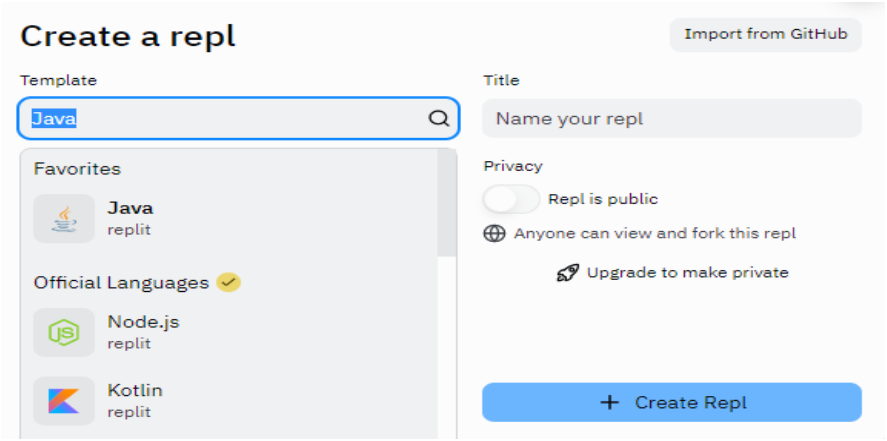
Step1: Creating a repl.it account

a) In your browser, visit https://repl.it
b) Clickon **Signup**
c) On a Sign-up page that appears, enter username, email and password
d) Click Sign up button to complete the registration

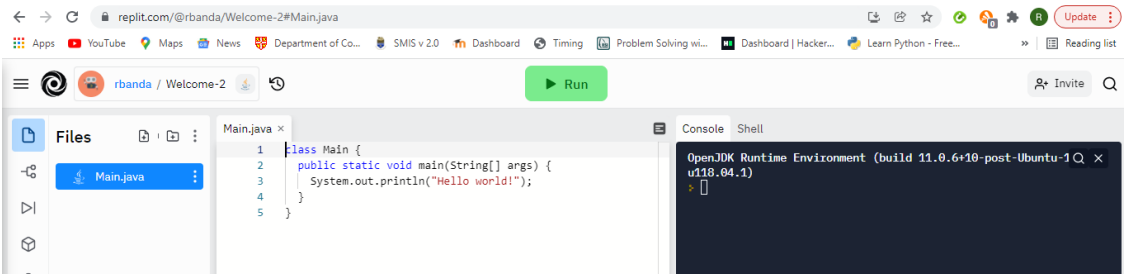
Step2: Creating the first Java repl



a) Log in to your repl.it account using the link https://repl.it/login
b) Click on **+New repl** button to create a repl. The window below appears

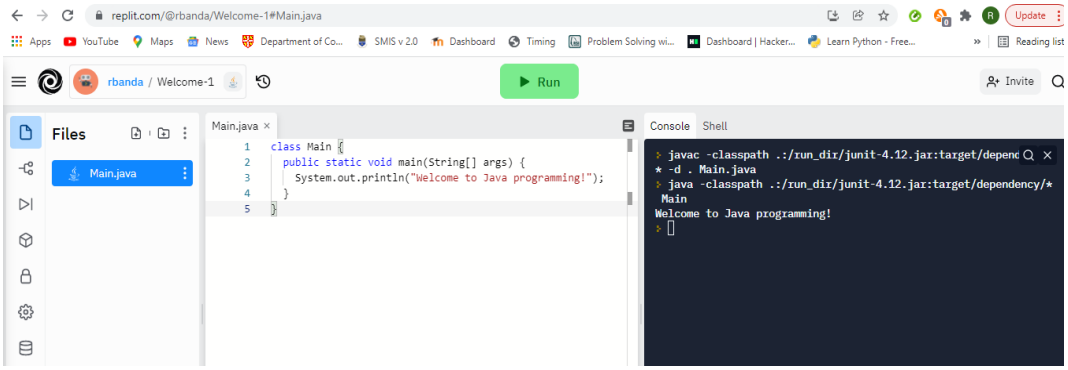


c) Search your language, i.e. Java.
d) Then name your repl as **Welcome**
e) Click **Create repl**. The browser IDE loads



e) Change the text, **"Hello world!"** to **"Welcome to Java Programming"**

f) Click the Run button to compile and run the program. The output will be as shown below.



LAB3: Using IDEs of your choice

In this lab, student is asked to explore different IDEs for Java and install the IDE of their choice aftermaking a properreviewofthe IDEs.

Prerequisites: A laptop or desktop with Operating System of your choice.

Step1: Review the three IDEs: NetBeans, Eclipse and IntelliJIDEA

a) Go on internet and search NetBeans vs Eclipse vs IntelliJ IDEA.
b) Select any result of your choice to learn based on how the three IDEs compare.

Step2: Install IDE of your choice (optional)

a) You can download and install the IDE of your choice which can be used for the labs in the subsequent chapters.

Please note that most of the labs in this module can be done without using a complicated IDEs listed above. However, mastering an IDE will increase your productivity.
