COMP5112 Lab 1

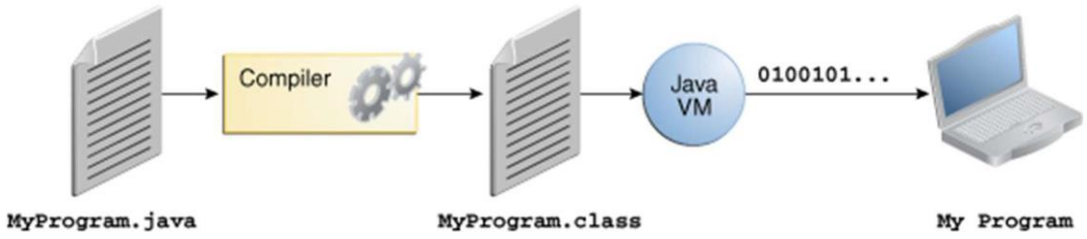
1. Objectives:

a) Learn the basic knowledge of how to use Java programming language.
b) Learn to install Java working environment.
c) Learn to create a Java program to output “Hello world!” on your computer.

2. Introduction:

Java was developed by James Gosling at Sun Microsystems Inc in 1995. It is a simple programming language which is easy to write, compile and debug. It helps to create reusable code and modular programs across different operating systems (write once and run anywhere).

Java’s syntax is similar to C/C++, so you can easily learn Java if you have some C/C++ programming experience. If you don't have any programming experience, you don't need to worry because after this lab, you will learn the basic operation of Java.



The figure above shows the procedure how to use Java to create a runnable program on your computer. It mainly contains three steps:

1. **Write your code in `.java` extension file.**
   All Java source code is first written in plain text files ending with the `.java` extension, which means you can write code in any text editor like notepad, and save the file with `.java` extension name.
2. **Compile your code using the `javac` compiler.**
   You will get the `javac` compiler after you have installed Java Development Kit on your computer. The compiler transforms the code (`.java` file) to a runnable program (`.class` file). Obviously, before generate the program, the compiler would check whether your code has any error, and it would try its best to tell you the reasons and locations of the found errors.
3. **Run `.class` file on Java VM.**
   The `.class` file cannot be directly understood by your operating system (e.g., running a .exe file on Windows is direct); it contains bytecodes which can be understood by Java Virtual Machine (Java VM), so you need to install Java VM on your computer to run Java program. The `java` launcher tool can run your application with an instance of the Java

VM.

In this lab, you will be hands-on with the development process of a Java program. You will firstly install Java Development Kit (JDK) on your computer. JDK contains `javac` compiler and `java` launcher tool which are mentioned above. Then, you will write a short and simple piece of Java code which tends to output “Hello World!”. At last, you need to compile and run your code. If you have extra time, you can try to modify your code and get familiar with Java syntax.

3. Sample Java Programs:

In Java code, any text that appears after // (in a line), or between /* and */ (could be multiple lines), are comments, used to explain the code and make it easier to understand. They do not affect the functionality at all.

Here are two programs we will use in this lab.

The content of `HelloWorldApp.java` is:

```
/**
  * The HelloWorldApp class implements an application that
  * simply prints "Hello World!" to standard output.
  */
class HelloWorldApp {
        public static void main(String[] args) {
                System.out.println("Hello World!"); // Display the string.
        }
}
```

The content of `InputTest.java` is:

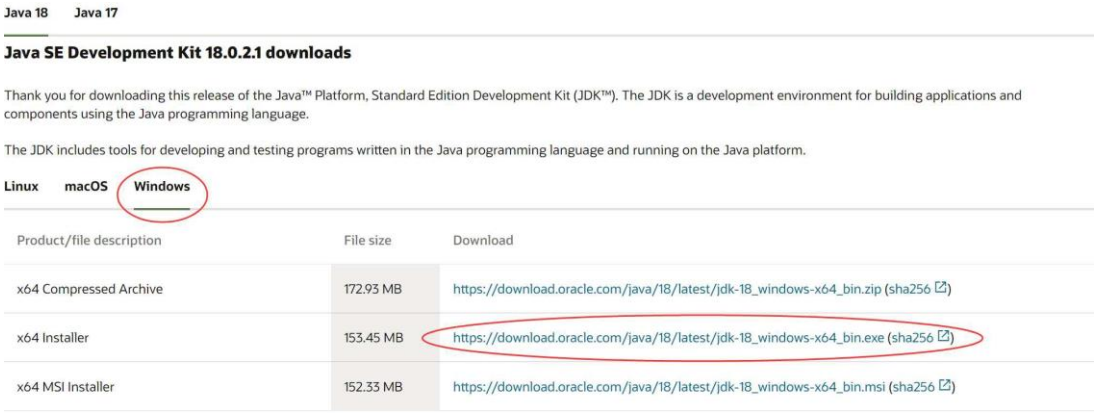
```
/**
 * The InputTest class implments an application that
 * read one integer from standard input (command line),
 * and then print "Your input integer is: xxx", xxx is
 * the read integer.
 */
// To use class from the library, you need to use the import keyword.
// Here we import java.util.Scanner for reading input.
import java.util.*;
class InputTest {
    public static void main(String[] args) {
         Scanner in = new Scanner(System.in); // Create a Scanner object named in.
        int a = in.nextInt(); // let in read a integer from command line and assign the value to variable a.
        System.out.println("Your input integer is: " + a);  // Display "Your input integer is: " and variable
a.
    }
```

}

4. Procedures:

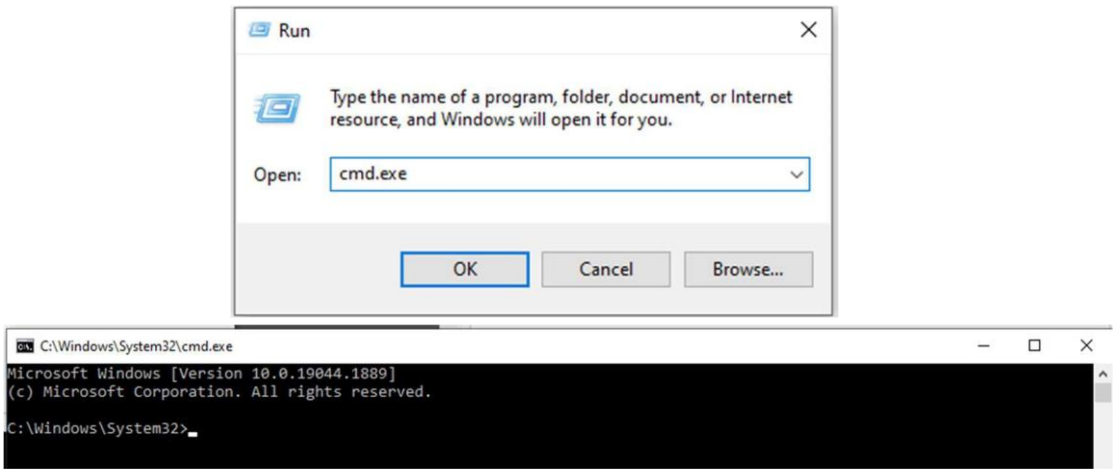
If you encounter problems in the following steps, you can view this page about common problems and solutions (https://docs.oracle.com/javase/tutorial/getStarted/problems/index.html).

4.1 **If your machine has installed JDK, you can skip 3.1 and 3.2.** Start your PC in PQ604 A, B, C and download Java SE Development Kit (JDK) for the corresponding operating system. The PCs in our labs use Windows system, so please download the Windows version.
JDK download link: https://www.oracle.com/java/technologies/downloads/
(or JDK21 is recommended. Alternative you may try higher version of JDK)

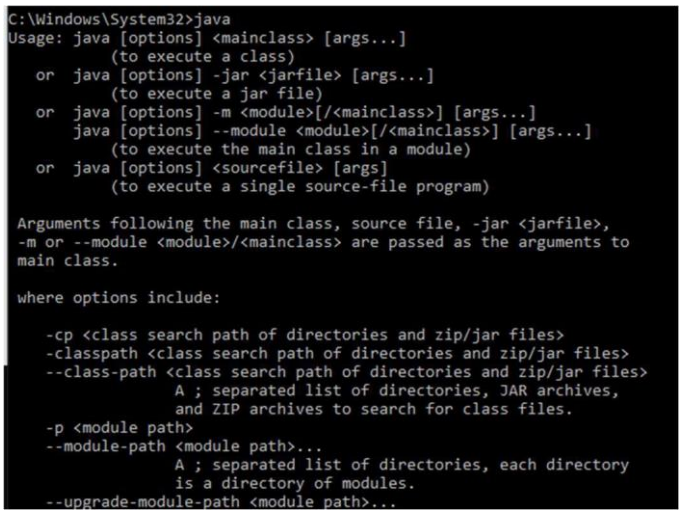


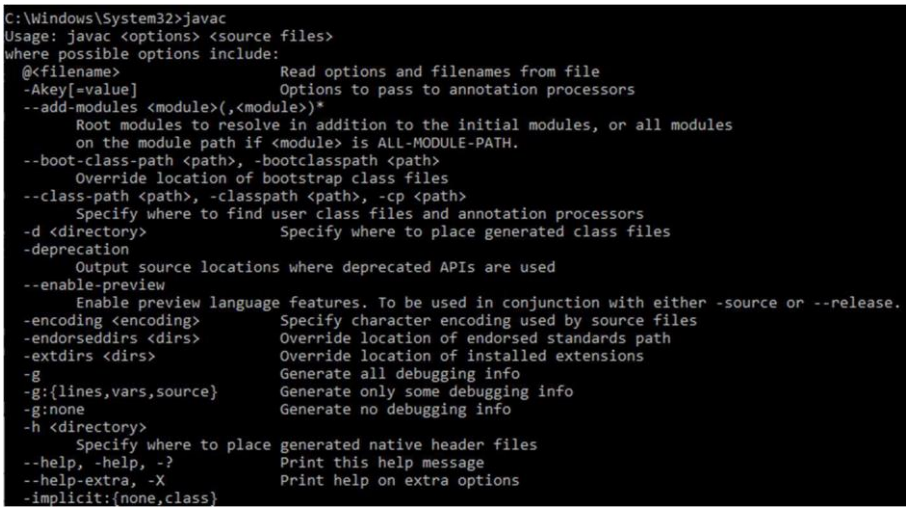
4.2 Install JDK on your computer after you have downloaded the Installer.

4.3 Check whether `javac` compiler and `java` launcher tool are correctly installed. To do this, firstly you need to bring up a command window. You can do this from the windows **Start** menu by choosing **Run...** and then entering `cmd.exe`. The command window should look similar to the following figure.



Then, type `java` and `javac` and press **Enter** key to see whether they work. If they are correctly installed, you will see your command returns a lot of description text on how to use `java` and `javac`, like the follow figures.





If your command returns something like "`'java' is not recognized as an internal or external command, operable program or batch file.`" That means JDK is **not** correctly installed or configured on your computer. Please check the above steps again.

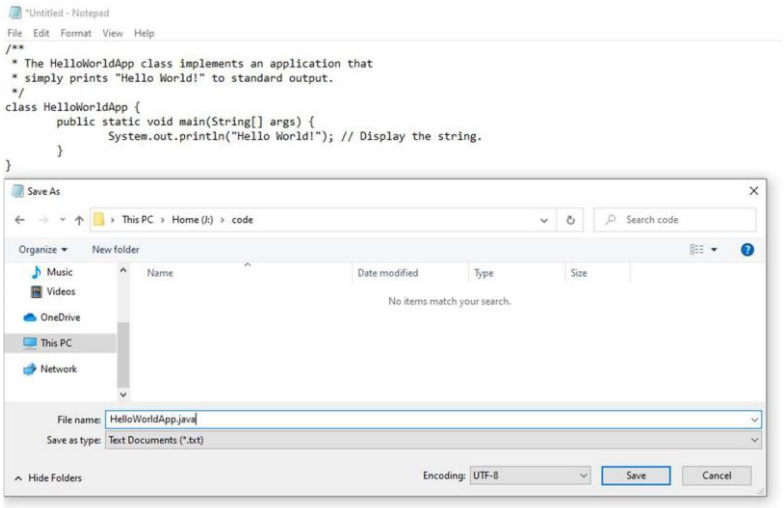
Note: Please notice that all code, commands, and file names are case-sensitive and should be type exactly as shown, so you must capitalize consistently. For example, `java` is different from `JAVA`, and `HelloWorldApp` is not the same as `helloworldapp`.

4.4 Create a `.java` source code file. First, start your text editor. You can launch the Notepad (or Notepad++, vscode, etc.) from the **Start** menu by selecting **Programs > Accessories > Notepad**. In a new document, type in the following code:

```
/**
  * The HelloWorldApp class implements an application that
  * simply prints "Hello World!" to standard output.
  */
class HelloWorldApp {
        public static void main(String[] args) {
```

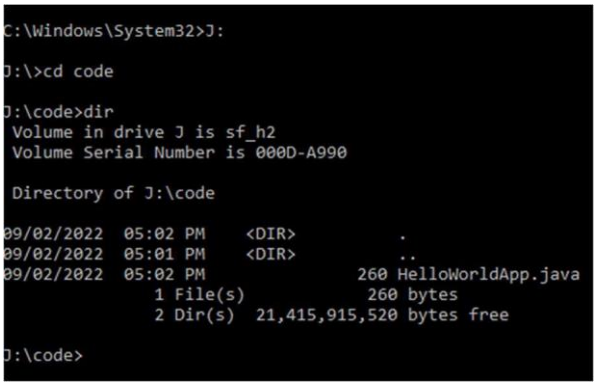
```
        System.out.println("Hello World!"); // Display the string.
    }
}
```

Save the code in a file with the name `HelloWorldApp.java`. To do this in Notepad, first choose the **File > Save As ...** menu item. Type the correct file name with `.java` extension name, click **Save**, and exit Notepad.



Note: If you don't like using notepad to write code, you are free to use any other text editor (Sublime, Atom, VS Code, Vim, Notepad++, etc.), or some more specialized software designed specifically for code editing (like IntelliJ IDEA and Eclipse). A better code editor can greatly improve the efficiency of writing code and prevent writing bugs.

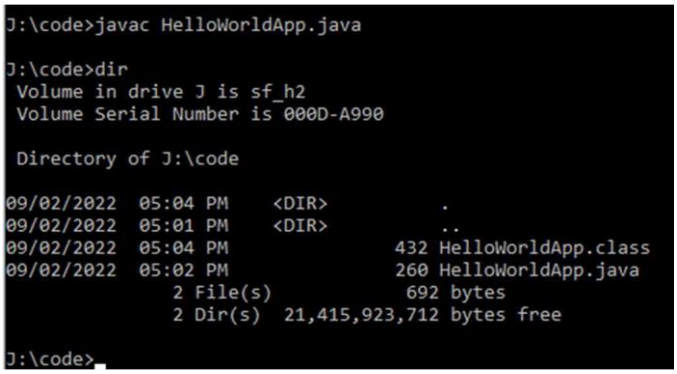
4.5 Compile the Source File into a `.class` File. Firstly, you need to bring up a command window (cmd.exe) same as above. Then you need to change your current directory to where `HelloWorldApp.java` is located.



For example, I have just saved `HelloWorldApp.java` in `J:\code`. Currently the prompt shows the current directory is `C:\Windows\System32`. So, I need to firstly enter `J:` to move to a different drive. Then, I need to enter `cd J:\code` or `cd code` to get to the target directory. You can use command `dir` to see the files in the current directory. Now you are ready to compile. At the prompt, enter the following command.

```
javac HelloWorldApp.java
```

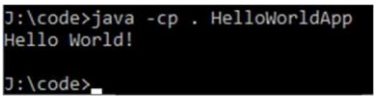
The compiler has generated a bytecode file, `HelloWorldApp.class`. At the prompt, type `dir` to see the new file that was generated as follows. Now that you have a `.class` file, you can run your program.



4.6 Run the Program. In the same directory, enter the following command at the prompt:

```
java -cp . HelloWorldApp
```

If you should see the following on your screen:

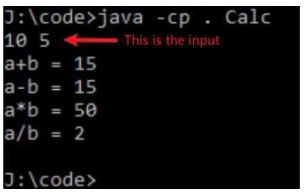


Congratulations! Your program works!

If you want to know how the code in `HelloWorldApp.java` works, you can check this page for a detailed explanation of the code.
https://docs.oracle.com/javase/tutorial/getStarted/application/index.html

4.7 If you have completed all the steps above, you can learn more about Java language syntax. You can refer to the following official tutorials or any other tutorials you like.
https://docs.oracle.com/javase/tutorial/java/nutsandbolts/index.html

Mini-challenge 1: please try to write a program named `Calc`, which can output the sum, difference, product, and quotient of two input integer `a` and `b`.



Hint: Here is a sample program to simply read one integer and output that integer, and

you can learn how to input and output in this program. Please note that the .java code file name must be the same as the class name, so you need to create `InputTest.java` if you want to run the following code.

```
/**
 * The InputTest class implments an application that
 * read one integer from standard input (command line),
 * and then print "Your input integer is: xxx", xxx is
 * the read integer.
 */
// To use class from the library, you need to use the import keyword.
// Here we import java.util.Scanner for reading input.
import java.util.*;
class InputTest {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in); // Create a Scanner object named in.
        int a = in.nextInt(); // let in read a integer from command line and assign the value to variable a.
        System.out.println("Your input integer is: " + a);  // Display "Your input integer is: " and variable
a.
    }
}
```

Mini-challenge 2: Please try to write a program named `Stars`, which read an input integer `n`, and then output `n` lines of star notations (*). Each output line has two more star notations than the previous line. You can implement this program after class. It is recommended to learn the loop statements first.

Mini-challenge 3: Please try to write a program named `StarsAndDollars`. This is very similar to the previous one. But this time, you are required to output star notations on odd-numbered line (line 1, line 3, line 5…) and output dollar notations on even-numbered line (line 2, line 4, line 6…). You can implement this program after class. It is recommended to learn the if-then statements first.

5. References:

[1] "About the Java Technology (The Java™ Tutorials)", *Docs.oracle.com*, 2022. [Online]. Available: https://docs.oracle.com/javase/tutorial/getStarted/intro/definition.html.

[2] ""Hello World!" for Microsoft Windows (The Java™ Tutorials)", *Docs.oracle.com*, 2022.

[Online]. Available: https://docs.oracle.com/javase/tutorial/getStarted/cupojava/win32.html.
[3]"Lesson: Language Basics (The Java™ Tutorials)", Docs.oracle.com, 2022. [Online]. Available: https://docs.oracle.com/javase/tutorial/java/nutsandbolts/index.html.