Lab 01

Java Installation, Compiling, Executing, Errors, and Testing

Objective:

The objective of this lab is to give students handon JAVA Installation/JAVA IDE, translating algorithms in Java program and execute it after correcting errors in it.

Activity Outcomes:

On completion of this lab student will be able

- Install JAVA SDK/ JAVA IDE
- Compile a program
- Execute JAVA program using JAVA SDK/JAVA IDE
- Test a program
- Debug program with syntax and logic errors.

Instructor Note:

As a pre-lab activity, read Chapter 01 from the text book "Java How to Program, Deitel, P. & Deitel, H., Prentice Hall, 2019".

1) Useful Concepts

This tutorial is for students who are currently taking a programming fundamentals course.

Introduction to Java

Java is a powerful and versatile programming language, developed by James Gosling in 1995, for developing software running on mobile devices, desktop computers, and servers. It is popular because of its unique feature of writing a program once and run it anywhere.

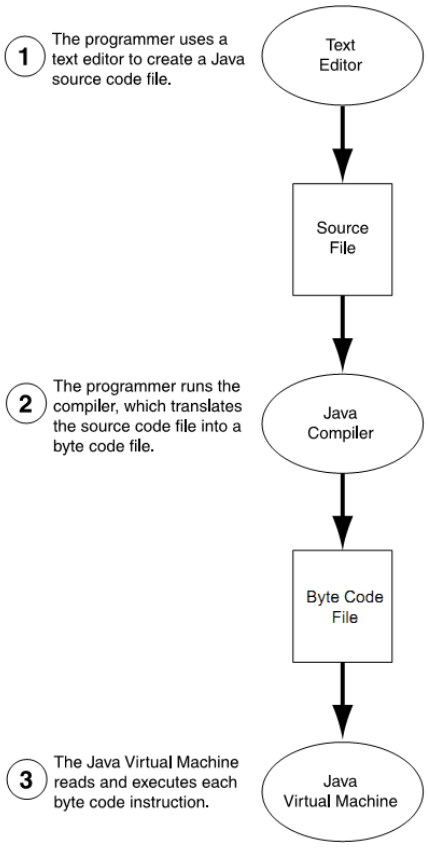
JDK Versions

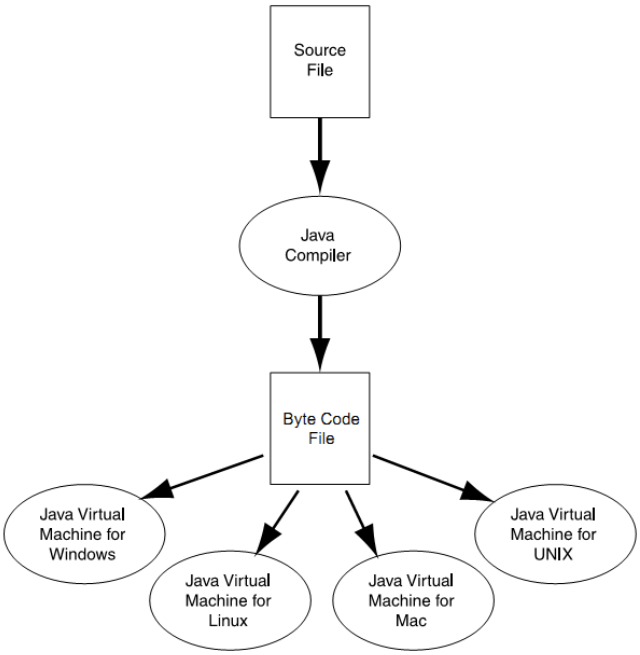
You can use the JDK command line utility to write Java programs. The JDK command line utility consists of a set of separate programs, such as compiler and interpreter, each of which is invoked from a command line. Besides the JDK command line utility, there are more than a dozen Java development tools on the market today, including IntelliJ, NetBeans, JBuilder, and Eclipse. These tools support an integrated development environment (IDE) for rapidly developing Java programs. Editing, compiling, building, debugging, and online help are integrated in one graphical user interface. Using these tools effectively will greatly increase your programming productivity.

The compiler and the Java Virtual Machine

The compiler is the program that translates source code into a language that a computer can understand. This process is different in Java than some other high-level languages. Java translates its source code into byte-code using the javac command. Students should be exposed to how the javac command gets executed in the particular development environment. Also, since javac is a program that can be run at the command prompt if using a Unix environment.

The compiler generates byte-code that is then interpreted by the Java Virtual Machine. This process occurs when the student uses the java command to execute the program. Once again, how this command is executed will vary depending on computing environments, and java is also a program that can be run at the command prompt. The interpretation of the byte-code by the Java Virtual Machine is the reason Java is considered so portable. The byte-code that is generated by the Java compiler is always the same no matter what the machine or operating system. As long as the Java Virtual Machine is loaded onto a computer, that computer can interpret the byte-code generated by the compiler. This second computer can be a different type or even running a different operating system than the one that originally compiled the source code and the byte-code will still be correctly interpreted





2) Solved Lab Activites

Sr.No	*Allocated Time*	*Level of Complexity*	*CLO Mapping*
Activity 1	***15 mins***	***Low***	***CLO-5***
Activity 2	***15 mins***	***Low***	***CLO-5***
Activity 3	***15 mins***	***Low***	***CLO-5***
Activity 4	***15 mins***	***Low***	***CLO-5***

Activity 1:

This activity demonstrate the steps to be followed to install Java on the system

Solution:

The JDK can be installed on the following platforms:

1- Microsoft Windows
2- Linux
3- macOS

Step-1: Download the latest JDK for Windows

You can download the JDK from Java SE Development Kit Downloads

Step-2: Running the JDK Installer

1- Start the JDK 23 installer (or latest) by double-clicking the installer's icon or file name in the download location.
2- Follow the instructions provided by the installer.
3- After the installation is complete, delete the downloaded file to recover the disk space.

Step-3: Checking if it is installed

To check if Java is installed (open command prompt and type following command)

```
java –version
```

Activity 2:

Setting Java Path (for using notepad, sublimeText or other text editor)

Solution:

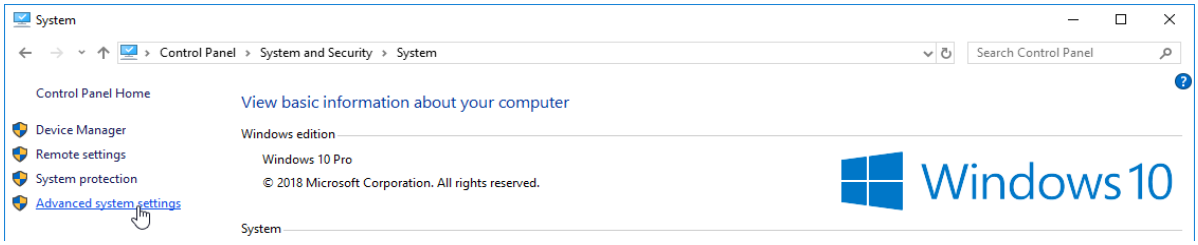
Temporary Path

1- Open the command prompt
2- Copy the path of the *JDK/bin* directory
3- Write in command prompt: *set path=copied_path*

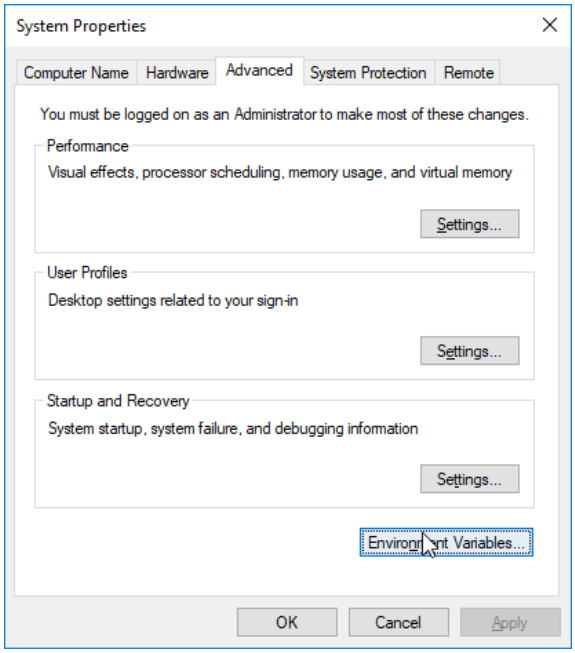
```
C:\Users\Your Name>set path=C:\Program Files\Java\jdk\bin
```

Permanent Path using Environment Variables Settings

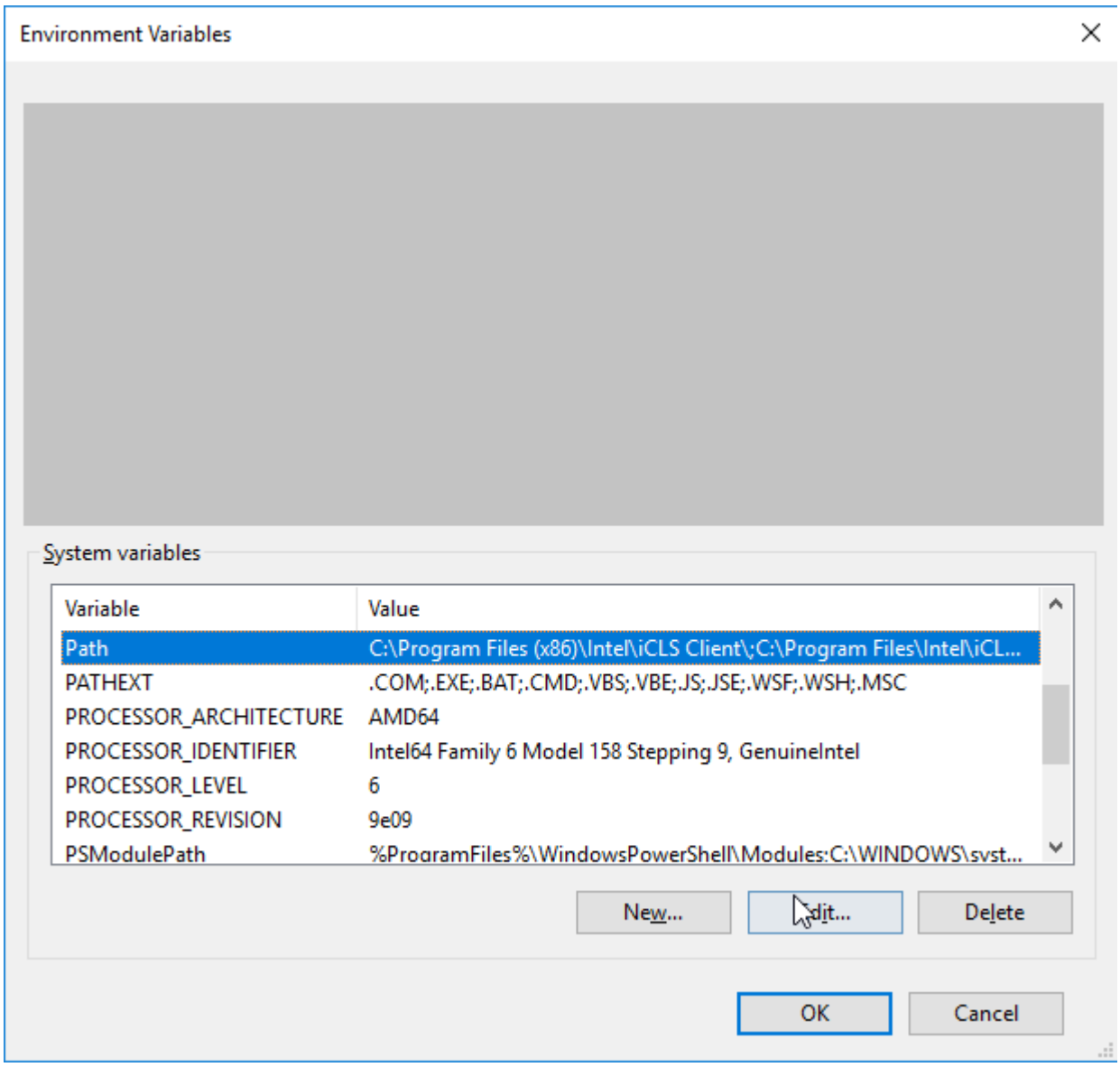
1- Go to "System Properties" (Can be found on Control Panel > System and Security > System > Advanced System Settings)



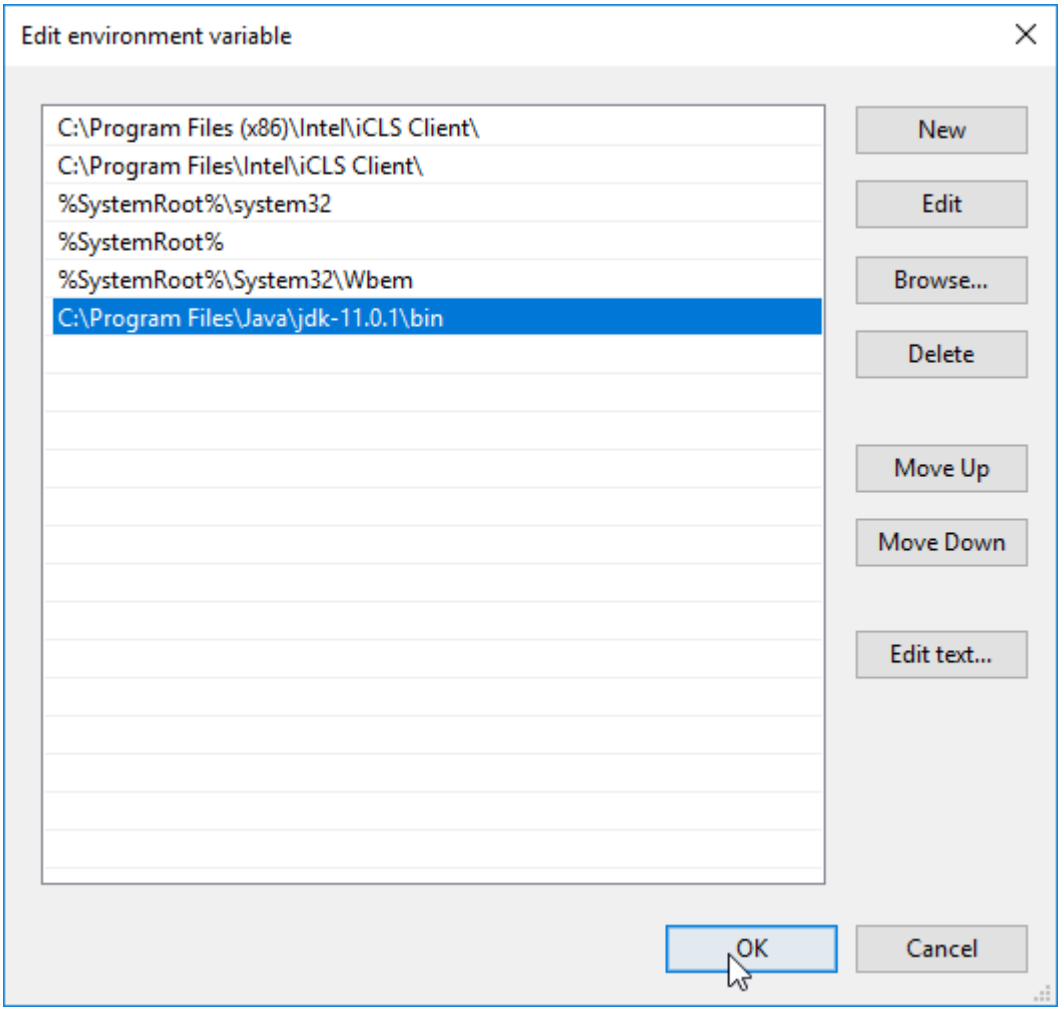
2- Click on the "Environment variables" button under the "Advanced" tab



3- Then, select the "Path" variable in *System variables* and click on the "*Edit*" button



4- Click on the "*New*" button and add the path where Java is installed, followed by \bin. By default, Java is installed in C:\Program Files\Java\jdk-15\bin



5- Then, click "OK", and save the settings
6- Write the following in the command line (cmd.exe):

```
C:\Users\Your Name>java -version
```

If Java was successfully installed, you will see something like this (depending on version):

```
java version "11.0.1" 2018-10-16 LTS
Java(TM) SE Runtime Environment 18.9 (build 11.0.1+13-LTS)
Java HotSpot(TM) 64-Bit Server VM 18.9 (build 11.0.1+13-LTS, mixed
mode)
```

Activity 3:

Writing, Compiling and Executing a JAVA program

Solution:

1. Open any text editor (notepad, sublimeText or notepad++) of your choice and write following code in editor

```
//This program calculates the user's gross pay

import java.util.Scanner;    //to be able to read from the keyboard

public class Pay
{
      public static void main(String [] args)
      {
            //create a Scanner object to read from the keyboard
            Scanner keyboard = new Scanner(System.in);

            //identifier declarations
            double hours;       //number of hours worked
            double rate;        //hourly pay rate
            double pay;         //gross pay

            //display prompts and get input
            System.out.print("How many hours did you work? ");
            hours = keyboard.nextDouble();
            System.out.print("How much do you get paid per hour? ");
            rate = keyboard.nextDouble();

            //calculations
            if(hours <= 40)
                  pay = hours * rate;
            else
                  pay = (hours - 40) * (1.5 * rate) + 40 * rate;

            //display results
            System.out.println("You earned $" + pay);
      }
}
```

2. Save the file using same name as the class name i.e. *Pay.java*
3. After saving the program, go to your operating system's command prompt and change your current directory or folder to the one that contains the Java program you just created. Then use the following command to compile the program.

```
javac Pay.java
```

4. You should not receive any error messages.
5. **Execute the Program:** Now enter the following command to run the program.

```
java Pay
```

6. When this program is executed, it will ask the user for input. You should calculate several different cases by hand. Since there is a critical point at which the calculation changes, you

should test three different cases: the critical point, a number above the critical point, and a number below the critical point. You want to calculate by hand so that you can check the logic of the program. Fill in the chart below with your test cases and the result you get when calculating by hand.

7. Execute the program using your first set of data. Record your result. You will need to execute the program three times to test all your data. Note: you do not need to compile again. Once the program compiles correctly once, it can be executed many times. You only need to compile again if you make changes to the code.

Hours	Rate	Pay (hand calculated)	Pay (program result)

Activity 4:

Debugging a Java Program

Solution:

1) Open any text editor (notepad, sublimeText or notepad++) of your choice and write following code in editor

```
//This program calculates the total price which includes sales //tax

import java.util.Scanner;

public class SalesTax
{
      public static void main(String[] args)
      {
            //identifier declarations
            final double TAX_RATE = 0.055;
            double price;
            double tax
            double total;
            String item;

            //create a Scanner object to read from the keyboard
            Scanner keyboard = new Scanner(System.in);
```

```
        //display prompts and get input
        System.out.print("Item description:   ");
        item = keyboard.nextLine();
        System.out.print("Item price:   $");
        price = keyboard.nextDouble();

        //calculations
        tax = price + TAX_RATE;
        total = price * tax;

        //display results
        System.out.print(item + "       $");
        System.out.println(price);
        System.out.print("Tax          $");
        System.out.println(tax);
        System.out.print("Total       $");
        System.out.println(total);
    }
}
```

2) Save the file as *SalesTax.java*.
3) This is a simple Java program that contains errors. Compile the program. You should get a listing of syntax errors. Correct all the syntax errors, you may want to recompile after you fix some of the errors.
4) When all syntax errors are corrected, the program should compile. As in the previous exercise, you need to develop some test data. Use the chart below to record your test data and results when calculated by hand.
5) Execute the program using your test data and recording the results. If the output of the program is different from what you calculated, this usually indicates a logic error. Examine the program and correct logic error. Compile the program and execute using the test data again. Repeat until all output matches what is expected.

Item	Price	Tax	Total (calculated)	Total (output)

3) Graded Lab Tasks

Note: The instructor can design graded lab activities according to the level of difficulty and complexity of the solved lab activities. The lab tasks assigned by the instructor should be evaluated in the same lab.

Lab Task 1

Every student is required to make installation on his / her personal computer before next lab