Day-1: Java Introduction

Prerequisite

- Basic of any programming language

An Introduction to Java

- Author: James Gosling
- Organization: SUN (Stanford University Network) Microsystems since 2010 SUN Microsystems is subsidiary of Oracle Corporation)
- Initial name of this language was Oak (This name is inspired by an Oak tree that stood outside James gosling office but name oak was already registered with a company that that oak technologies) so it was renamed to java.
- It is a high level, object oriented programming language
- It is used to develop desktop, web and mobile applications

Java Edition

The following editions of java are important

1. Java Standard Edition (Java SE)

- It describes an abstract Java platform.
- It provides a foundation for building and deploying network-centric enterprise applications that range from the PC desktop computer to the workgroup server.
- It is implemented by the Java Software Development Kit (SDK). This edition is popularly known as **Core Java**.

2. Java Enterprise Edition (Java EE)

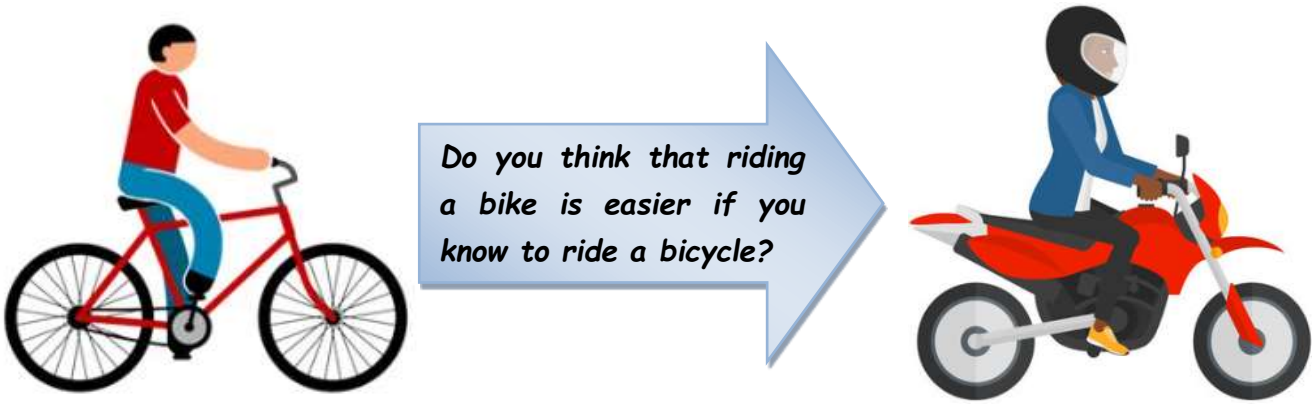
- Java Platform, Enterprise Edition (Java EE) is based on the Java SE specification.
- It is used for developing and running large-scale, multi-tiered, scalable, reliable and secure network applications

3. Java Micro Edition (Java ME)

- A robust, flexible environment for applications running on embedded and mobile devices
- Applications based on Java ME are portable across many devices, yet leverage each device's native capabilities.

Java Features

1. Simple

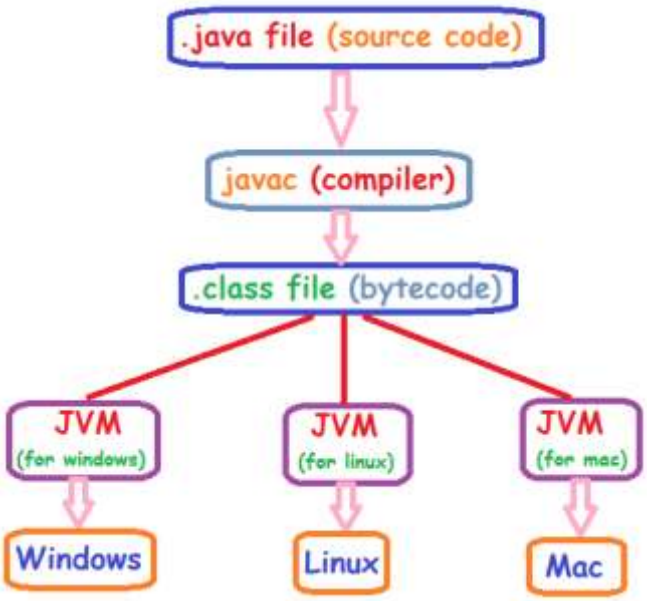


- Look at the timeline: C (1972) → C++ (1983) → Java (1992)
- Java inherits the C/C++ syntax and many of the object-oriented features of C++, also some of the more confusing concepts from C++ are either left out of Java or implemented in a cleaner, more approachable manner. So if you are familiar with C/C++ then it will take a little effort to learn java.
- A common source of complexity in many C and C++ applications is storage management. The allocation and freeing of memory, java provides automatic garbage collection, thereby simplifying the task of programmer.

2. Portable

An Analogy: *Let me give you a situation, we have a gadget repairing store where lot of different variety of devices are repaired; as you already know that every device is having specific type of charger (like some mobile phones used type-A cable, some uses type-C cable, printer and scanner uses type-B cable and the apple phones uses apple pin charger); for the store owner it is not possible to have separate charger for every gadgets so there is a solution that is called universal charger that has pins for every type of charger.*

java also uses the same kind of model for the to achieve architecture neutrality which becomes an indispensible in the era of internet.



- Whenever someone compiles a java program, its output is not an executable code but it is an intermediate highly optimized set of instruction that is executed by java runtime system, known as JVM (Java virtual machine), that intermediate set of instruction is called **Bytecode**.
- Bytecode runs on JVM so only the JVM needs to be implemented for each platform. Once the run-time package (that is JVM) exists for a given system, any Java program can run on it. Remember, although the details of the JVM will differ from platform to platform, all interpret the same Java bytecode. That feature make java an **Architecture neutral** language & JVM is referred as an **interpreter for bytecode**. Java works on the concept of **Write Once Run Anywhere**.
- There are no "implementation-dependent" notes in the Java language specification. C and C++ both suffer from the defect of designating many fundamental data types as "implementation dependent" while Java defining standard behaviour that will apply to the data types across all platforms. Java specifies the sizes of all its primitive data types and the behaviour of arithmetic on them.
- **Portability = Architecture Neutrality + Standard data type**

3. Interpreted

- Java bytecode are translated on the fly to native machine instructions (interpreted) and not stored anywhere. And linking is a more incremental and lightweight process, the development process can be much more rapid and exploratory.

4. Security

- Security manager is a component of JVM that is responsible for security policy of java. All the dynamic content that may harm the system is run under the sandbox model (kind of quarantined).
- Apart from the compile time checks, Java performs various run-time checks that can be done only at run time, hence improve security.
- Java has provides a separate middleware service in JAAS [java authentication and authorization service] which provide web security, Auth and SSO.

5. Object Oriented

- Java uses a formal OOP type system that must be obeyed at compile-time and run-time. This is helpful for larger projects, where the structure helps keep the various parts consistent

6. Robust

- Java is a strongly typed language, one of the advantages of a strongly typed language is that it allows extensive compile-time checking so bugs can be found early.
- Java puts a lot of emphasis on early checking for possible problems, later dynamic (runtime) checking, and eliminating situations that are error prone.
- Garbage collector is responsible for managing memory means leaking of memory due to mismanagement is avoided.
  Exception handling is another feature in Java that makes for more robust programs.

7. Multithreaded

- Java supports multithreaded programming, which allows you to write programs that do many things simultaneously.

We Activity

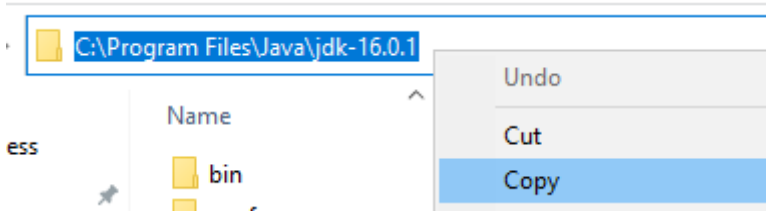
Downloading and installing java on Windows Machine

- Java can be downloaded from the link given below for windows operating systems

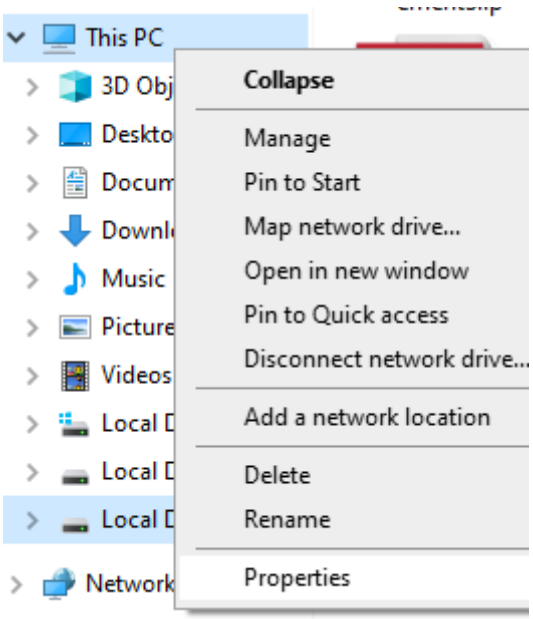
  https://www.oracle.com/in/java/technologies/downloads/

  Double click on the setup that you have downloaded from the link given above. After installation have completed, go through following steps:-

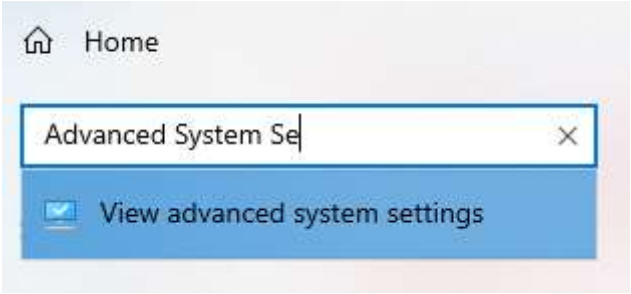
- Copy you java installation path up to bin directory (it will be something like “C:\Program Files\Java\jdk-16.0.1\bin”)



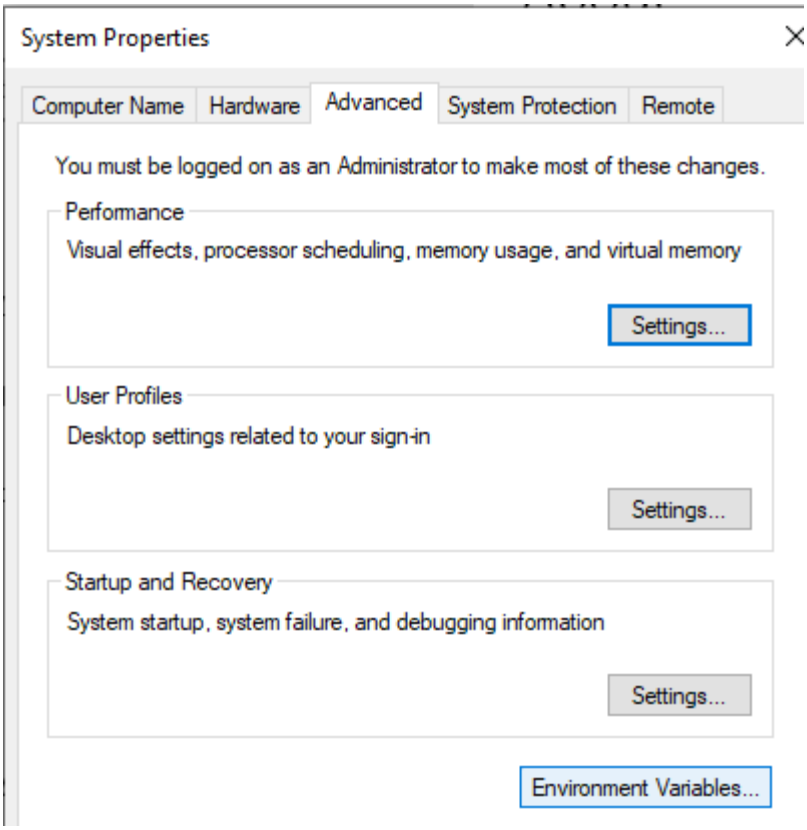
- Right Click on **This PC** icon provided on your desktop screen or in start menu.



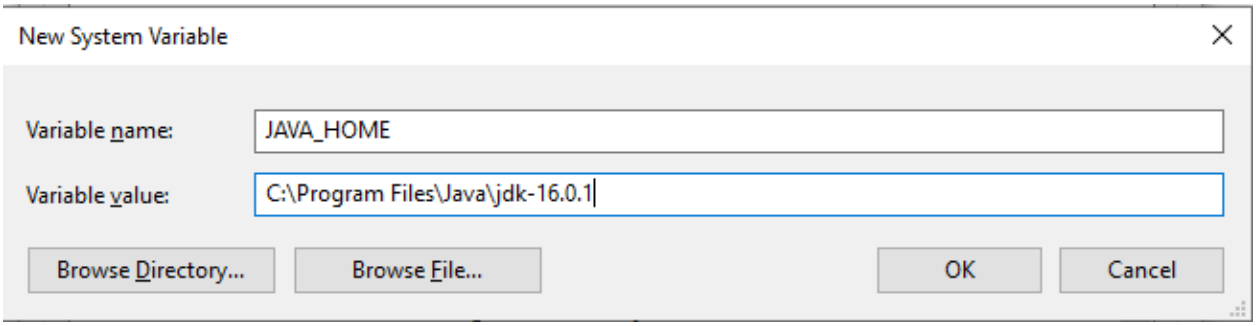
- After clicking on properties button you will get a new window, In that window search for “Advanced System Settings” then click on “View Advanced System Settings”



- Now click on the **Advanced** then click on the **Environment Variable**

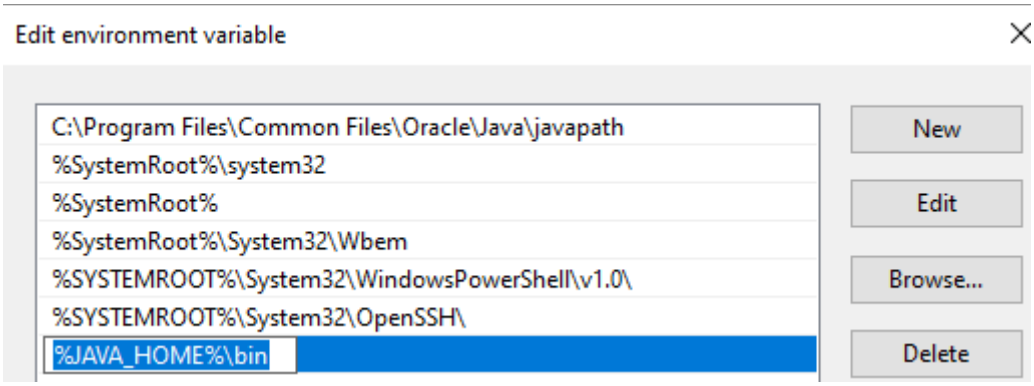


- Clicking will open another dialog box. In that newly opened dialog box you will find list of user variables as well as system variables. You will be able to edit system variable only if you are logged in as administrator. In the system variable click on the new button then another dialog box will appear and in the same enter variable name JAVA_HOME and paste the copies value for the variable value



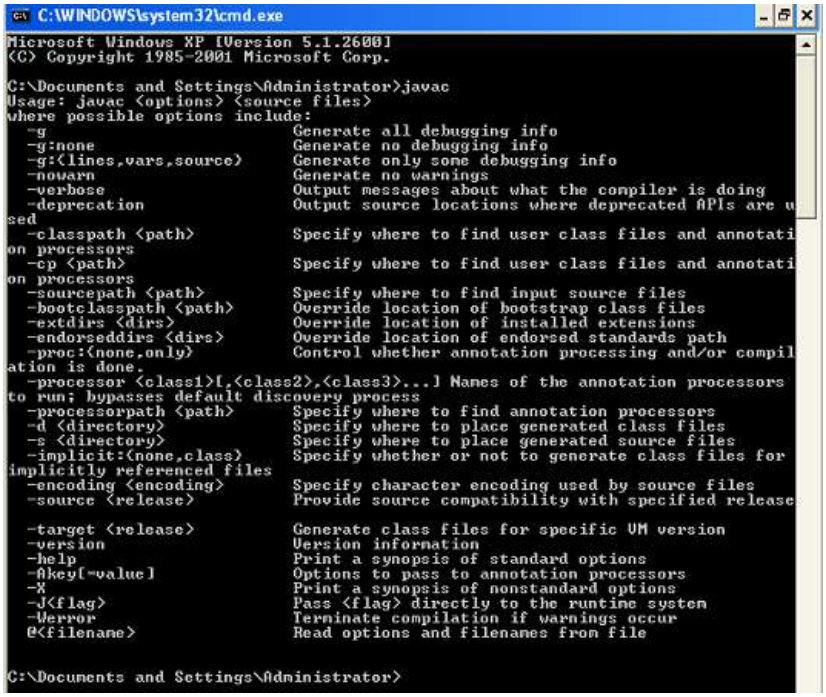
- Now click on the variable ‘PATH’ and then click on edit button; a new dialog box will apear, in the

the dialog box click on new and Write following value %JAVA_HOME%/bin



- Now click on ok buttons in all opened dialog boxes and the window opened by right clicking on This PC then properties.

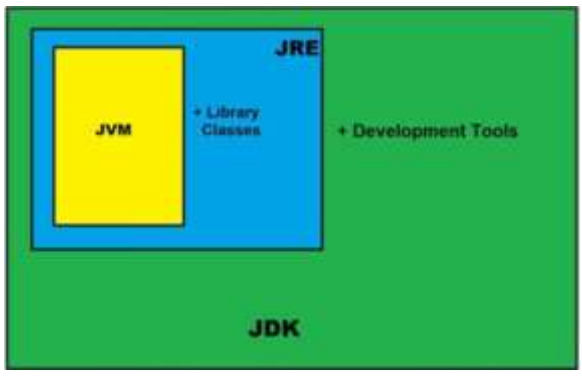
- Now go to start menu click on the run button type cmd in text box then click on run button, console will be appeared in front of you. Now type javac then press enter. you will get list of all options along with their description.



- Your java installation has been completed.

JDK and JRE

- JDK (Java Development Kit) is used when you want to develop and run the java code
- JRE (Java runtime environment) is used when you want to run the java code but not to develop the java code; Both JVM and JRE contains the JVM



I Activity

Writing First Java Program

Step-1: Create a folder name **com** and inside the com folder create another folder **masai**
Step 2: open notepad and save the file with name Main.java
Step 3: Now type following program in editor (notepad)

```
/*
      This is a simple java program to print Hello World
*/
package com.masai;

public class Main {
      public static void main(String args[]) {
            System.out.println("Hello world");
      }
}
```

Tip: Be careful while typing, java is a case sensitive language.

Step 4: Now open command prompt/terminal, move to the **com/masai**/ folder and type following command

```
javac Main.java
```

The javac compiler creates a file called Main.class that contains the bytecode version of the program. bytecode is the intermediate representation of your program that contains instructions the Java interpreter will execute. Thus, the output of javac is not code that can be directly executed. General syntax of compiling java program is

```
javac <filename with extension>
```

Step 5: and now come to the **parent folder of the com folder** and type following command

```
java com.masai.Main
```

You will get output

```
Hello world
```

When you execute the Java interpreter as just shown, you are actually specifying the name of the class that you want the interpreter to execute. General syntax is

```
java <fully-qualified-class-name that contains main method>
```

It will automatically search for a file by that name that has the .class extension. If it finds the file, it will execute the code contained in the specified class.

Explaining First Program

Our program begins with the line

```
/*
      This is a simple java program to print Hello World
*/
```

This is a comment. Like other programming languages, contents of a comment are ignored by the compiler. Instead, a comment describes or explains the operation of the program to anyone who is reading its source

code, then

```
package com.masai;
```

Like it is a common habit to keep the files in the folder in the same way in java it is common to keep classes in the interfaces. by this code we are keeping the class Main in package com.masai. rest of the details about the package we will do later.

```
public class Main{
```

This line uses the keyword class to declare that a new class is being defined. Main is an identifier that is the name of the class. In java programming all code has to be written in the class. (except for package and import statement)
Tip: If a class is declared with public access modifier then the name of file containing the class must has name same as of the class i.e. if a file can have only public class. If class is not public then filename and class has no relation.

```
public static void main(String args[]) {
```

This is the line at which the program will begin executing.

- public means that function is visible to all over.
- static means method main can be called using class-name. Here the position of static and public keyword can be changed
- void tell the compiler that main method will not return any value
- main is the name of the method
- The String args[ ] declares as a parameter named, which is an array of instances of the class String. you might choose other name instead of args. The details about the same will be discussed in the command line arguments. The same can be written as String[ ] args or String… args.

```
System.out.println("Hello world ");
```

- The System is inbuilt class
- The out is an object of PrintStream class such that the former is written inside the System class as a static member i.e. it is accessible using the class name.
- println() is a inbuilt method that output everything to console whatever is passed as a parameter.

This statement will simply print `"Hello world"` on console followed by new line.

Finally, two consecutive closing curly braces one for closing main method and another for closing class A.

Tip: Everything that you code in java is written within class including variable to methods (even main), that is why it is called object oriented language.

An Analogy

Installing spring tool suite (STS) for java code

STS Logo

- Download the STS for your operating system from https://spring.io/tools and make simple double click installation
- After installation, open the STS and for one time only, you need to select a folder as workspace (it is a folder where STS will put all code for you).
- To create a project, click on New -> Java project, a dialog box will be opened where you need to enter the project name (you may uncheck checkbox “create module-info.java” for now)and then click on Finish, you project will be listed in the package explorer.
- Right click on the src folder -> New -> package -> type package name com.masai and hit enter
- Right click on the com.masai -> New -> class -> type class name Main and hit enter
- in the body of class Main, just type main and hit ctrl + space; select the first suggestion you main method is written
- Inside the main method type sysout, and hit ctrl + space it will be expanded to System.out.println(); write “Hello world” inside the println(). you have done with your code.
- No need to write any command for running as STS is already compiling the code as you are typing the same
- To run the code, no need to write any command as you just need to right click in the editor → Run As → java Editor; or you may use the short key ALT + SHIFT + X, J. your program will start running.

The STS is an **IDE (integrated Learning Environment)** that allows writing, compiling, debugging and running java code.

Naming conventions for java

- **for packages:** In all-lowercase ASCII letters and should be one of the top-level domain names, currently com, edu, gov, mil, net, org or one of the English two-letter codes identifying countries. Subsequent components of the package name vary according to an organization's own internal naming conventions. Such conventions might specify that certain directory name components be division, department, project, machine, or login names.
  e.g. com.sun.eng, edu.cmu.cs.bovik.cheese, com.masai.curriculum

- **for class name and interface:** Should be nouns, in mixed case with the first letter of each internal word capitalized.
  e.g. SimpleInterestCalculator, ThreadGroup

- **for Methods:** should be verbs, in mixed case with the first letter lowercase, with the first letter of each internal word capitalized
  e.g. getInterestAmount(), runFast()

- **for variables:** All instance variables, class, and class constants are in mixed case with a lowercase first letter. Internal words start with capital letters. Variable names should be short yet meaningful. The choice of a variable name should be mnemonic- that is, designed to indicate to the casual observer the intent of its use.

e.g. principleAmount, length, totalBillAmount
Tip: One-character variable names should be avoided except for temporary "throwaway" variables. Common names for temporary variables are i, j, k, m, and n for integers; c, d, and e for characters.

- **for Constants:** The names of variables declared class constants and of ANSI constants should be all uppercase with words separated by underscores ("_"). (ANSI constants should be avoided, for ease of debugging.)
  e.g. MIN_WIDTH, CPU_CORES, MAX_PRIORITY

You Activity

Try to execute following code on your computer and then observer the difference between System.out.println() and System.out.print() and use of \n.

```
//filename: DiffPrintStatement.java
package com.masai;

public class DiffPrintStatement {
	static public void main(String... masai) {
		System.out.print(“Twinkle Twinkle ”);
		System.out.println(“Little Star”);
		System.out.print(“How I wonder what you are\n”);
		System.out.println(“Up above the world so high”);
		System.out.println(“Like a diamond in the sky”);
	}
}
```