What is Java? (Popular Perception)

Monday, October 19, 2020   11:36 AM

Popular Perception/Definition

- Java is an object oriented programming language
- Class based OO language
- High level programming language
- Multi-threaded

- Write once - run anywhere
- Cross platform language
- Platform Independent

- Robust and Secured
- Interpreted language

- It is an open source

Is this claim valid?

Contradictions/Challenges

- **Does Android Phone support Java?**
  - **NO**
  - A Java class doesn't run on Android Phones out of box
  - Although majority android application is written using java language.

- **Does iOS support Java?**
  - **NO**

- **Have you heard the term — "This computer doesn't have Java?"**
  - **We havent installed Java.**
  - **What have we not installed — Java Programming lanugage?**
  - **Do everyone needs to program in Java?**
- **When you try to download "java", what are your actually downloading?**

- Why request for a Java download ended up downloading JRE?
- Is JRE and object oriented programming language?
- Is JRE platform independent
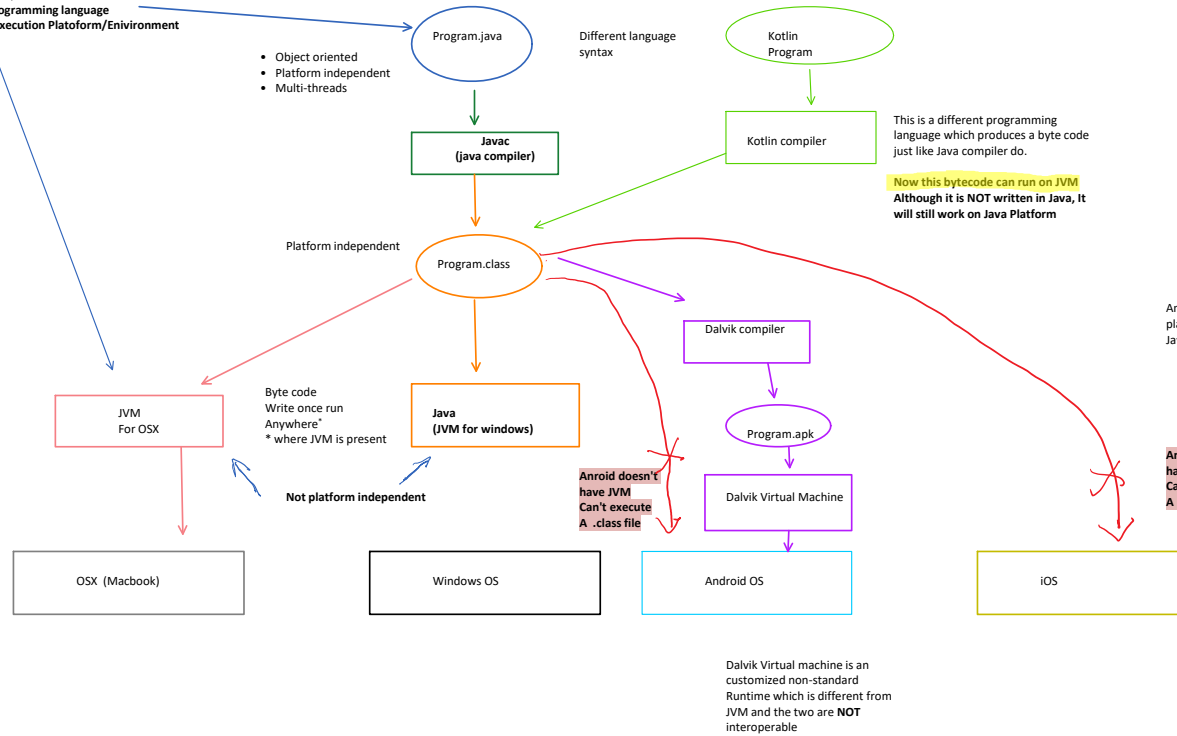- Can you write an object oriented programming language using JRE?

How many Java are there?

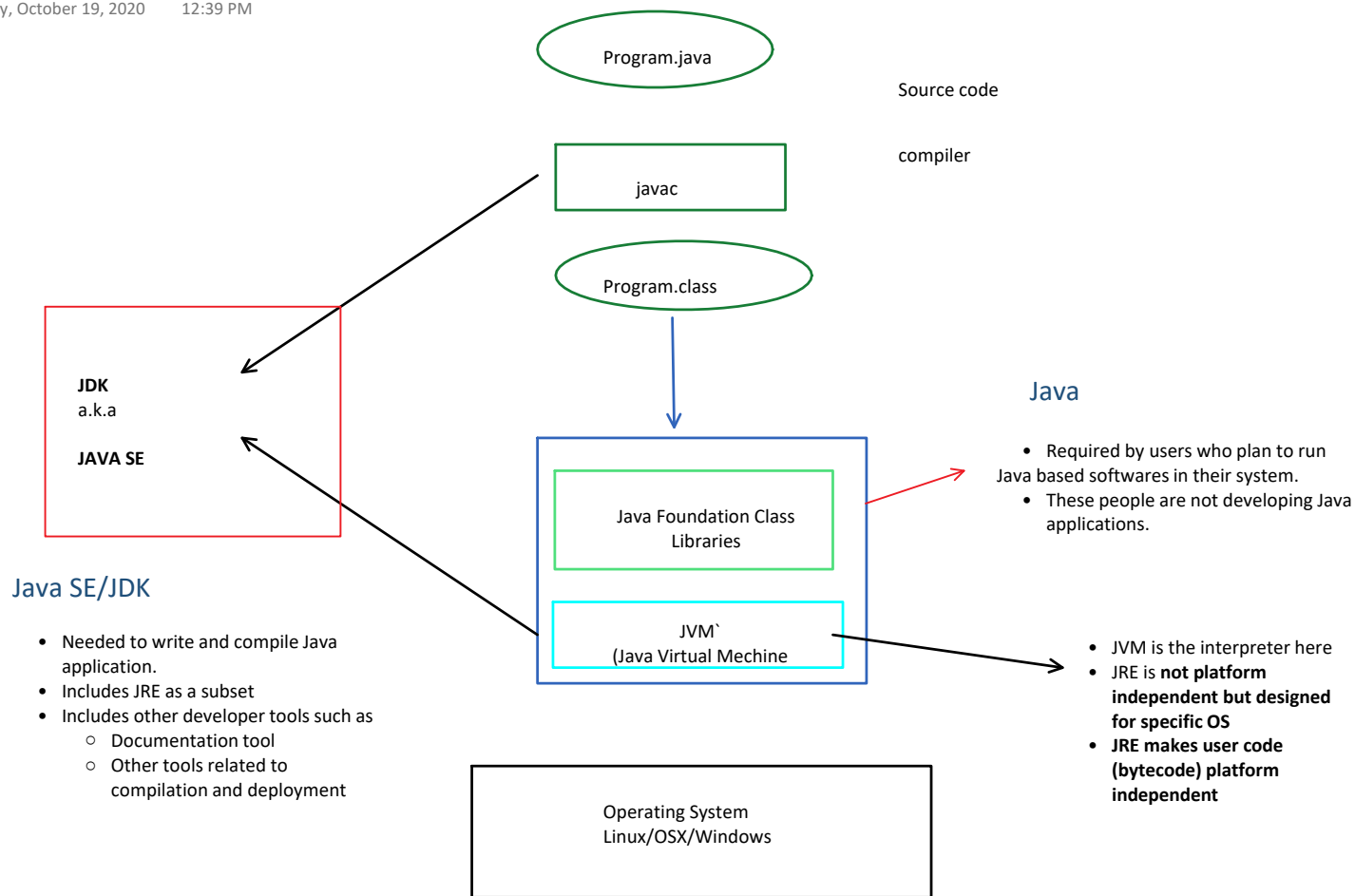In the world of computation **Java** refers to two related but **distinct terms**
1. **Java is a Programming language**
2. **Java is an execution Platoform/Enivironment**

Program.java

- Object oriented
- Platform independent
- Multi-threads

Different language syntax

Kotlin Program

Since the release of Android Studio 3.0 in October 2017, Kotlin is included as an alternative to the standard Java compiler. The Android Kotlin compiler lets the user choose between targeting Java 6 or Java 8 compatible bytecode. Kotlin has been Google's preferred language for Android app development since 7 May 2019.

Javac (java compiler)

Kotlin compiler

This is a different programming language which produces a byte code just like Java compiler do.

Now this bytecode can run on JVM
Although it is NOT written in Java, It will still work on Java Platform

Platform independent

Program.class

Dalvik compiler

Android and IOS doesn't support Java platform and they are not considered Java systems.

JVM For OSX

Byte code
Write once run Anywhere*
* where JVM is present

Java (JVM for windows)

Program.apk

Anroid doesn't have JVM Can't execute A .class file

Dalvik Virtual Machine

Anroid doesn't have JVM Can't execute A .class file

Not platform independent

OSX (Macbook)

Windows OS

Android OS

iOS

Dalvik Virtual machine is an customized non-standard Runtime which is different from JVM and the two are **NOT** interoperable

Java Framework

Program.java

Source code

javac

compiler

Program.class

JDK
a.k.a

JAVA SE

Java

- Required by users who plan to run Java based softwares in their system.
- These people are not developing Java applications.

Java Foundation Class Libraries

Java SE/JDK

- Needed to write and compile Java application.
- Includes JRE as a subset
- Includes other developer tools such as
  - Documentation tool
  - Other tools related to compilation and deployment

JVM`
(Java Virtual Mechine

- JVM is the interpreter here
- JRE is **not platform independent but designed for specific OS**
- **JRE makes user code (bytecode) platform independent**

Operating System
Linux/OSX/Windows

Java Environments based on Use cases

Server needs to install
JRE+ Web Runtime (like web server)

Just needs standard browser.
Doesn't require to install anything related to java
- NO JRE
- NO other software

Java desktop application user
- Needs to install JRE
- The product will include additional library

Webapp user

Java Desktop API

Java Web API

JDK

contains

JAVA   Runtime Environment

Contains JVM
Foundation class libraries
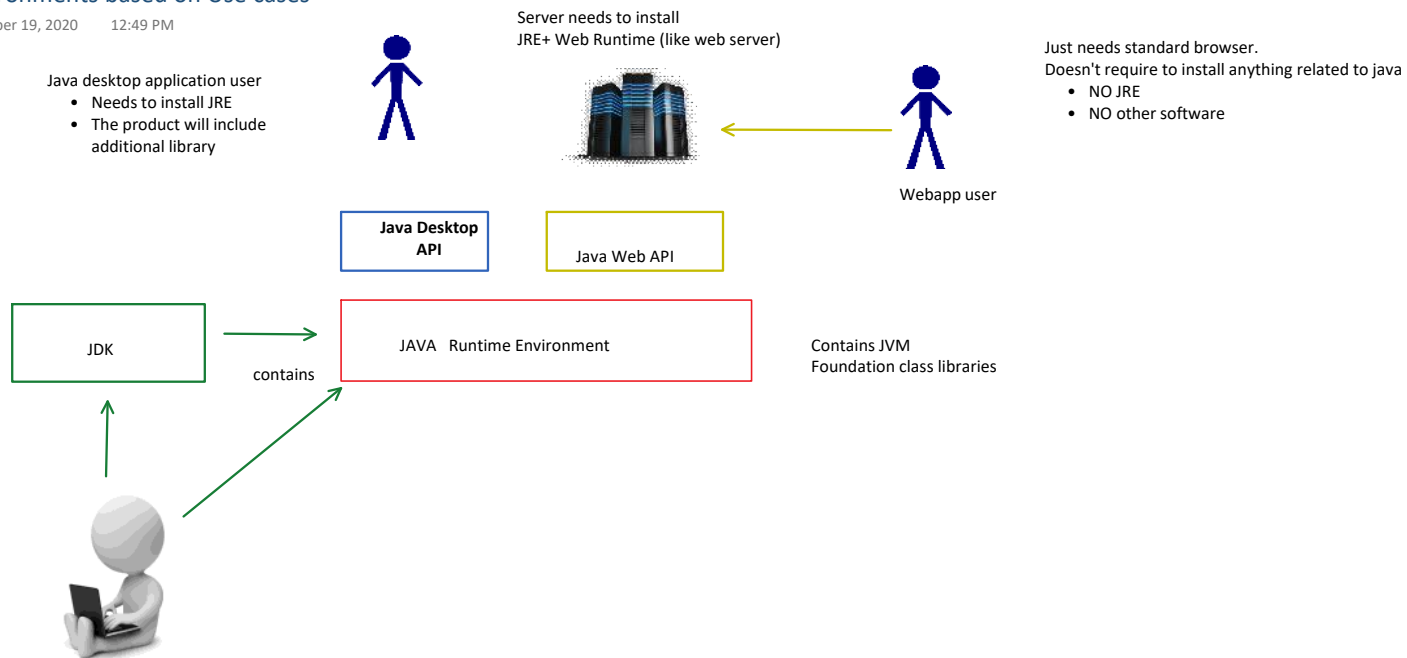
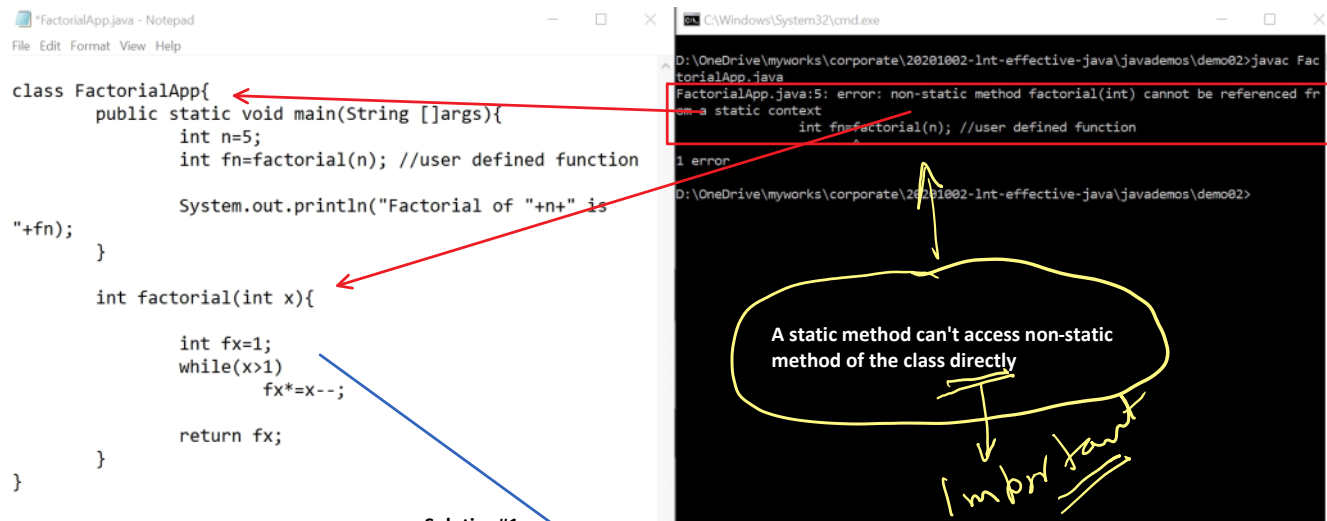# Compiling and Running Java code

Monday, October 19, 2020   1:16 PM
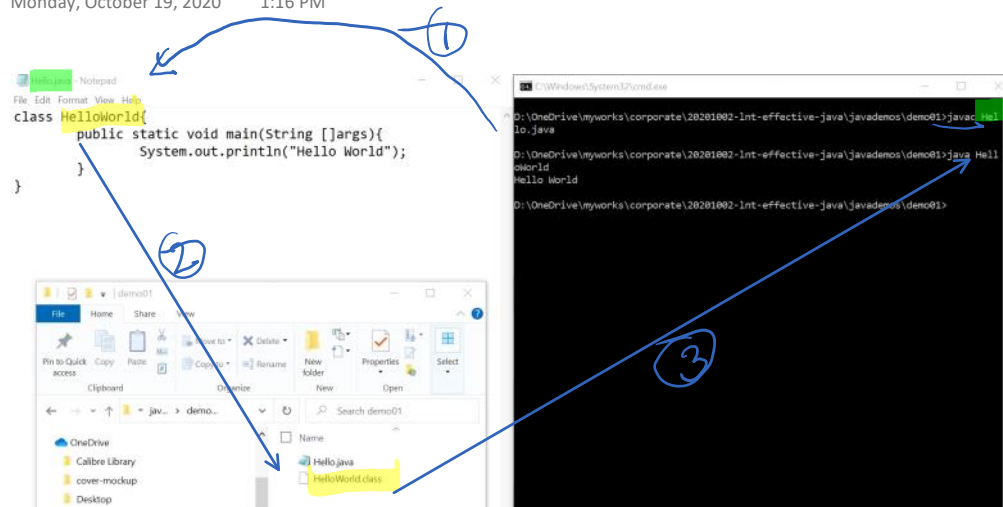
1. You compile a source file

`c:>` `javac` `Hello.java`

   1. The byte code name is same as that of Class present in source code and not same as the .java file name
   2. You run the byte code using java command

`c:>` `java` `HelloWorld`

Static context

```
class FactorialApp{
        public static void main(String []args){
                int n=5;
                int fn=factorial(n); //user defined function

                System.out.println("Factorial of "+n+" is
"+fn);
        }

        int factorial(int x){

                int fx=1;
                while(x>1)
                        fx*=x--;

                return fx;
        }
}
```

```
D:\OneDrive\myworks\corporate\20201002-1nt-effective-java\javademos\demo02>javac Fac
torialApp.java
FactorialApp.java:5: error: non-static method factorial(int) cannot be referenced fr
om a static context
                int fn=factorial(n); //user defined function

1 error

D:\OneDrive\myworks\corporate\20201002-1nt-effective-java\javademos\demo02>
```

A static method can't access non-static method of the class directly

Important

Solution#1

Solution
- There are multiple way to solve this problem
- We will choose different solutions depending on the context
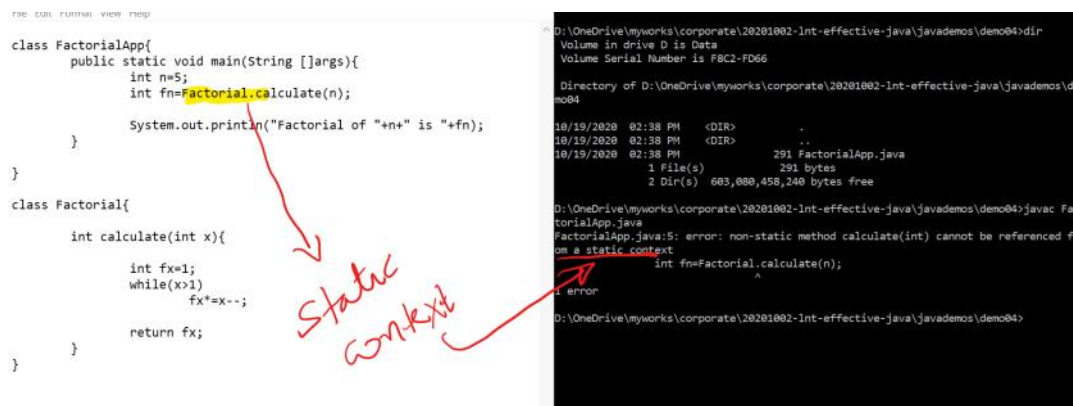
Solution 1
- Make factorial function static

```
static  int factorial(int x){

        int fx=1;
        while(x>1)
                fx*=x--;

        return fx;
}
```

More on static context

```
class FactorialApp{
        public static void main(String []args){
                int n=5;
                int fn=Factorial.calculate(n);

                System.out.println("Factorial of "+n+" is "+fn);
        }

}

class Factorial{

        int calculate(int x){

                int fx=1;
                while(x>1)
                        fx*=x--;

                return fx;
        }
}
```

Static context

```
D:\OneDrive\myworks\corporate\20201002-1nt-effective-java\javademos\demo04>dir
 Volume in drive D is Data
 Volume Serial Number is F8C2-FD66

 Directory of D:\OneDrive\myworks\corporate\20201002-1nt-effective-java\javademos\de
mo04

10/19/2020  02:38 PM    <DIR>          .
10/19/2020  02:38 PM    <DIR>          ..
10/19/2020  02:38 PM               291 FactorialApp.java
              1 File(s)            291 bytes
              2 Dir(s)  603,080,458,240 bytes free

D:\OneDrive\myworks\corporate\20201002-1nt-effective-java\javademos\demo04>javac Fac
torialApp.java
FactorialApp.java:5: error: non-static method calculate(int) cannot be referenced fr
om a static context
                int fn=Factorial.calculate(n);
                                ^
1 error

D:\OneDrive\myworks\corporate\20201002-1nt-effective-java\javademos\demo04>
```

A Class name is a static context. You can call only static methods using Class reference and not non-static methods

Working with a non-static context

To work with a non-static method, we need an object of the class to

Working with a non-static context

To work with a non-static method, we need an object of the class to refer and use the method.


```
class FactorialApp{
        public static void main(String []args){
                int n=5;

                //step#1:get object of the class
                Factorial fact=new Factorial();

                //step#2: call method using the object
                int fn=fact.calculate(n);

                System.out.println("Factorial of "+n+" is "+fn);
        }
}

class Factorial{

        int calculate(int x){

                int fx=1;
                while(x>1)
                        fx*=x--;

                return fx;
        }
}
```

Non Static → instance of object level Method

```
D:\OneDrive\myworks\corporate\20201002-lnt-effective-java\javademos\demo04>dir
 Volume in drive D is Data
 Volume Serial Number is F8C2-FD66

 Directory of D:\OneDrive\myworks\corporate\20201002-lnt-effective-java\javademos\demo04

10/19/2020  02:38 PM    <DIR>          .
10/19/2020  02:38 PM    <DIR>          ..
10/19/2020  02:44 PM               370 FactorialApp.java
               1 File(s)            370 bytes
               2 Dir(s)  603,080,458,240 bytes free

D:\OneDrive\myworks\corporate\20201002-lnt-effective-java\javademos\demo04>javac FactorialApp.java

D:\OneDrive\myworks\corporate\20201002-lnt-effective-java\javademos\demo04>java FactorialApp
Factorial of 5 is 120

D:\OneDrive\myworks\corporate\20201002-lnt-effective-java\javademos\demo04>dir
 Volume in drive D is Data
 Volume Serial Number is F8C2-FD66

 Directory of D:\OneDrive\myworks\corporate\20201002-lnt-effective-java\javademos\demo04

10/19/2020  02:45 PM    <DIR>          .
10/19/2020  02:45 PM    <DIR>          ..
10/19/2020  02:45 PM               312 Factorial.class
10/19/2020  02:45 PM               943 FactorialApp.class
10/19/2020  02:45 PM               399 FactorialApp.java
               3 File(s)          1,654 bytes
               2 Dir(s)  603,080,450,048 bytes free

D:\OneDrive\myworks\corporate\20201002-lnt-effective-java\javademos\demo04>
```

Multi Class Java Program

Monday, October 19, 2020   2:36 PM

FactorialApp.java - Notepad

File Edit Format View Help

```java
class FactorialApp{
        public static void main(String []args){
                int n=5;
                int fn=Factorial.calculate(n);

                System.out.println("Factorial of "+n+" is "+fn);
        }
}

class Factorial{

        static  int calculate(int x){

                int fx=1;
                while(x>1)
                        fx*=x--;

                return fx;
        }
}
```

When compiled a separate .class is generated for every class that exists in our system

C:\Windows\System32\cmd.exe

```
D:\OneDrive\myworks\corporate\20201002-lnt-effective-java\javademos\demo03>dir
 Volume in drive D is Data
 Volume Serial Number is F8C2-FD66

 Directory of D:\OneDrive\myworks\corporate\20201002-lnt-effective-java\javademos\de
mo03

10/19/2020  02:34 PM    <DIR>          .
10/19/2020  02:34 PM    <DIR>          ..
10/19/2020  02:34 PM               291 FactorialApp.java
               1 File(s)            291 bytes
               2 Dir(s)  603,080,462,336 bytes free

D:\OneDrive\myworks\corporate\20201002-lnt-effective-java\javademos\demo03>javac Fac
torialApp.java

D:\OneDrive\myworks\corporate\20201002-lnt-effective-java\javademos\demo03>java Fact
orialApp
Factorial of 5 is 120

D:\OneDrive\myworks\corporate\20201002-lnt-effective-java\javademos\demo03>dir
 Volume in drive D is Data
 Volume Serial Number is F8C2-FD66

 Directory of D:\OneDrive\myworks\corporate\20201002-lnt-effective-java\javademos\de
mo03

10/19/2020  02:35 PM    <DIR>          .
10/19/2020  02:35 PM    <DIR>          ..
10/19/2020  02:35 PM               312 Factorial.class
10/19/2020  02:35 PM               925 FactorialApp.class
10/19/2020  02:34 PM               291 FactorialApp.java
               3 File(s)          1,528 bytes
               2 Dir(s)  603,080,458,240 bytes free

D:\OneDrive\myworks\corporate\20201002-lnt-effective-java\javademos\demo03>
```

Class Dependencies
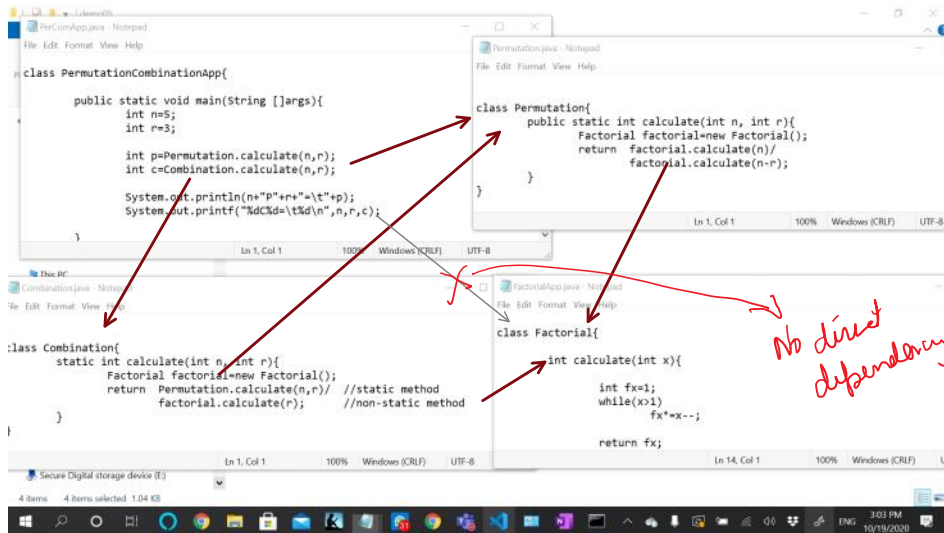
- Unlike c/c++ (or event javascript or python)
You don't need any kind of include of classes before you can use.
  - An .class present in the current folder can be accessed directly.

How to compile a project with multiple files

Method #1 compile all files using * wildcard

```
c:> javac *.java
```

Method #2 compile the startup file — the one that contains main()

The steps would be as follows

1. It will try to compile PerComApp.java to create class PermutationCombinationApp.class.
   - While compiling it needs to use Permutation class
2. Javac searches for a file Permutation.class
   - It is not currently present
3. Javac searches for file Permutation.java.
   - It compiles Permutation.java to create Permutation.class
4. While compiling Permutation.class, it realizes it Needs Factorial.class
   - There is no Factorial.class present
5. Since there is not Factorial.class Present, it searches for Factorial.java
   - Factorial.java is also not present
     - Factorial class is a part of FactorialApp.java file. It is not housed in a file name of its own
   - **At this stage it returns an error message**

if I have both Combination.class and Combination.java file present, which of them will be used by javac?

- If both files are present, then javac would compare their last modification data.
- If .class file is more recent than .java file, that means there has been no change in source code since last compilation, it would simply use the class file.
- If the class file is modified after last compilation, that means we must rebuild class file
  - It would recompile the java file

Recommendation!

- We should create one class per java file
- A class should be housed in a java file of same name
- This will help Java compiler automatically compile your java file if required.

Compilation based on Modification

Name	Status	Date modified	Type	Size
Combination.class	⊘	10/19/2020 3:27 PM	CLASS File	1 KB
PermutationCombinationApp.class	⊘	10/19/2020 3:27 PM	CLASS File	2 KB
Combination.java	⊘	10/19/2020 3:26 PM	JAVA File	1 KB
Factorial.class	⊘	10/19/2020 3:20 PM	CLASS File	1 KB
Permutation.class	⊘	10/19/2020 3:20 PM	CLASS File	1 KB
Permutation.java	⊘	10/19/2020 3:19 PM	JAVA File	1 KB
FactorialApp.java	⊘	10/19/2020 3:18 PM	JAVA File	1 KB
Factorial.java	⊘	10/19/2020 3:17 PM	JAVA File	1 KB
PerComApp.java	⊘	10/19/2020 2:56 PM	JAVA File	1 KB

Combination has changed after its last compile
So it is recompiled along with PerComApp.java

Factorial class has not changed after its
Last modification so it is not recompiled

```
D:\OneDrive\myworks\corporate\20201002-lnt-effective-java\javademos\demo05>javac PerComApp.java

D:\OneDrive\myworks\corporate\20201002-lnt-effective-java\javademos\demo05>
```

This file will always be compiled irrespective of its date
as we are explicitly compiling

Organization Project files in multiple folders

Monday, October 19, 2020   3:32 PM

Java compiler by default doesn't know where to search for the java/class files if it is not present int current folder.

It doesn't search entire file system for those files

How do I locate .java/.class files

1. CLASSPATH
- We can specify an environment variable called **CLASSPATH** listing all the folders where javac/java should search for .java/.class files
- Folders should be separated using path separator character that varies from one os to another
  - Windows using semicolon
  - Linux/osx uses colon

Our class path should look like

```
c:>set classpath=.\lib\console;.\lib\maths;.
```

Current directory

Note
- We have mentioned 3 path here
  - .\lib\console
  - .\lib\maths
  - . (current directory)
- We can't specify just lib
  - We must include right sub directory
- If you are having a classpath, then java/javac doesn't search current directory by default.
  - You must include current directory if you have files in current directory

```
      C:\Windows\System32\cmd.exe                                                    —   □   ×
Volume serial number is F8C2-FD66
D:.
    FactorialApp.class
    FactorialApp.java
    PerComApp.java                                         Java is not searching
    PermutationCombinationApp.class                        for .class file in the current
                                                           folder because current
    └──lib                                                 folder is not present in
        ├──console                                         class path
            ConsoleWriter.class
            ConsoleWriter.java

        └──maths
            Combination.class
            Combination.java
            Factorial.class
            Factorial.java
            Permutation.class
            Permutation.java

D:\OneDrive\myworks\corporate\20201002-1nt-effective-java\javademos\demo06>java PermutationCombinationApp
Error: Could not find or load main class PermutationCombinationApp
Caused by: java.lang.ClassNotFoundException: PermutationCombinationApp

D:\OneDrive\myworks\corporate\20201002-1nt-effective-java\javademos\demo06>
```

Works correctly with the Right Path Set


```
      C:\Windows\System32\cmd.exe                                                —   □   ×
D:\OneDrive\myworks\corporate\20201002-1nt-effective-java\javademos\demo06>set classpath=.\lib\console;.\lib\maths;.

D:\OneDrive\myworks\corporate\20201002-1nt-effective-java\javademos\demo06>tree /f
Folder PATH listing for volume Data
Volume serial number is F8C2-FD66
D:.
    FactorialApp.class
    FactorialApp.java
    PerComApp.java
    PermutationCombinationApp.class         Works fine with the right classpath

    └──lib
        ├──console
            ConsoleWriter.class
            ConsoleWriter.java

        └──maths
            Combination.class
            Combination.java
            Factorial.class
            Factorial.java
            Permutation.class
            Permutation.java

D:\OneDrive\myworks\corporate\20201002-1nt-effective-java\javademos\demo06>java PermutationCombinationApp
5P3=    60
5C3=    10

D:\OneDrive\myworks\corporate\20201002-1nt-effective-java\javademos\demo06>
```

Note About class path

- Any classpath set at the terminal or the command window is good for current session only and is lost once you close the window.
- Classpath or any environment variable set on a terminal/command window is not available to other terminal or the command window.
- If you need a classpath everyday then you must store it in **system environment variables**
  - **A good place to store classpath for common libraries.**

```
C:\Windows\System32\cmd.exe                                                            —    □

Microsoft Windows [Version 10.0.19041.508]
(c) 2020 Microsoft Corporation. All rights reserved.

D:\OneDrive\myworks\corporate\20201002-lnt-effective-java\javademos\demo06>java PermutationCombinationApp
5P3=    60
5C3=    10

D:\OneDrive\myworks\corporate\20201002-lnt-effective-java\javademos\demo06>cd \

D:\>java PermutationCombinationApp
5P3=    60
5C3=    10

D:\>
```

==Once a classpath is properly set, you can run your application from anywhere in your file system.==

Assignment01

Create A Project called Furniture App with following file structure

```
<project root>
      |
      |—> FurnitureApp.java
      |
      |——> [lib]
            |
            |——>[furnitures]
                    |
                    |——> Chair.java
                    |——> Bed.java

          |——>[data]
                  |
                  |——> List.java
```
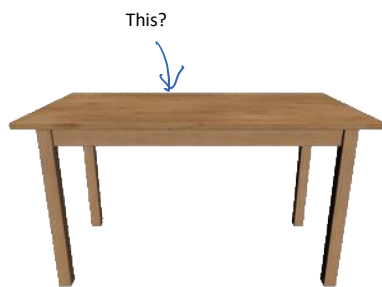
FurnitureApp should
1. Craete a list of Furnitures
2. Add Chair and Bed to this list

Write necessary code to compile and run the program

What is a Table?

This?

That?

Product	Rate	Stock
Chair	1000	12
Table	5000	7
Bed	20000	4

- In real world a word can have different meanings. Same word (like Table) can represent multiple different and unrelated elements.
- Often in a single project we may need to use one or more such objects
  - Example:
    - A Furntiure Shop sells **Table (Furnitures)**
    - The maintain their stock details in a **Table (Data)**

How do I represent Multiple Objects with same name in same application

### Why Multiple Folder Design Doesn't work?

- Java/Javac searches for java classes in **sequential order** in every folder mentioned in the **classpath**
- They stop searching when they find their **first match**
- If a class with same name is present in the two folders, it will always match the first folder mentioned in the list and will never use other option.

If we change the order in classpath, it will get a different Table. But we can't get Both Tables to work in single Application
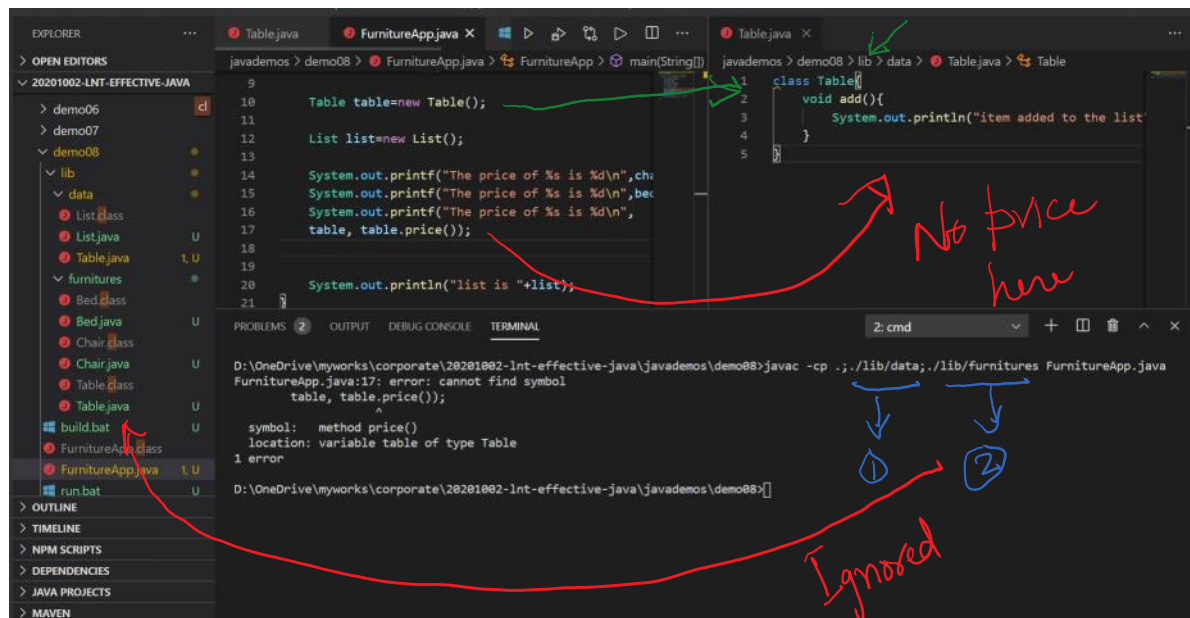
How to make it work?

- Create classes with different Names —
  - FurntiureTable
  - DataTable
- This can avoid name conflicts
- **Why this is not a great idea**
  - We may not always be in a position to find a good prefix
  - Sometimes conflict may be between
    - Your Data Table
    - My Data Table
  - Two different developers may be developing a class for same Purpose
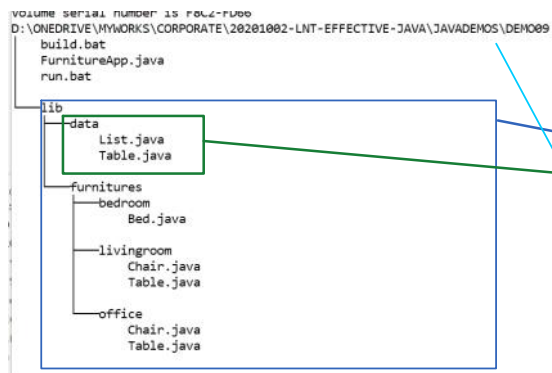   - Prefixing won't be useful here!

Java Package

- A java package is a logical grouping for your Java classes and packages
- A Package can contain
  - Java files
  - Class files
  - Other Resource (configurations)
  - Sub Packages
- **A package is physically mapped to a folder on the disk.**
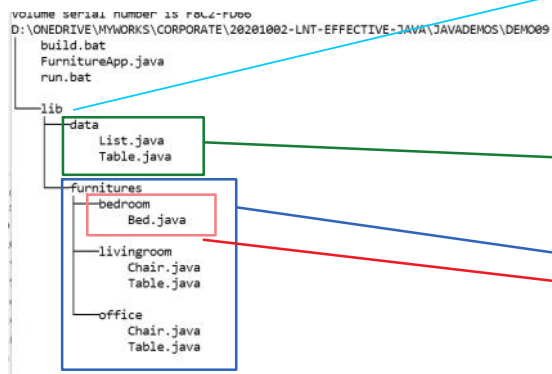- For every package you will have folder

Package is not a folder

- Package resides in a folder.
  - If we have a package **xyz** it would be mapped to a folder **xyz**
- **The key difference between a package and a folder**
  - Folder is an OS concept and Java doesn't know about the folder
    i. You reach the folders using OS level environment variable like **classpath**
    ii. Java program internally knows nothing about a folder

  - Package is a java concept mapped to folders.
    - **It is a java programming element and using in Java Program**
  - **We can designate our selected folder as package**
    - **That would be our choice**

```
volume serial number is F8C2-FD66
D:\ONEDRIVE\MYWORKS\CORPORATE\20201002-LNT-EFFECTIVE-JAVA\JAVADEMOS\DEMO09
    build.bat
    FurnitureApp.java
    run.bat
    lib
        data
            List.java
            Table.java
        furnitures
            bedroom
                Bed.java
            livingroom
                Chair.java
                Table.java
            office
                Chair.java
                Table.java
```

Package	Sub Package	Sub Package	Sub Package	Class	Java Name	
lib					**lib**	
	data				**lib.data**	
				List	lib.data.List	
				Table	lib.data.Table	
	furniture				**lib.furntiures**	
		bedroom			**lib.furnitures.bedroom**	
				Bed	lib.furnitures.bedroom.Bed	

Alternative View

```
volume serial number is F8C2-FD66
D:\ONEDRIVE\MYWORKS\CORPORATE\20201002-LNT-EFFECTIVE-JAVA\JAVADEMOS\DEMO09
    build.bat
    FurnitureApp.java
    run.bat
    lib
        data
            List.java
            Table.java
        furnitures
            bedroom
                Bed.java
            livingroom
                Chair.java
                Table.java
            office
                Chair.java
                Table.java
```

- This is just a folder and not a package.
- **This is** where the you package lives.
- **This folder should be present in the classpath**
- Your package folders will not be present in classpath

Package	Sub Package	Sub Package	Class	Java Name	
data				**data**	
			List	data.List	
			Table	data.Table	
furniture				**furntiures**	
	bedroom			**furnitures.bedroom**	
			Bed	furnitures.bedroom.Bed	