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

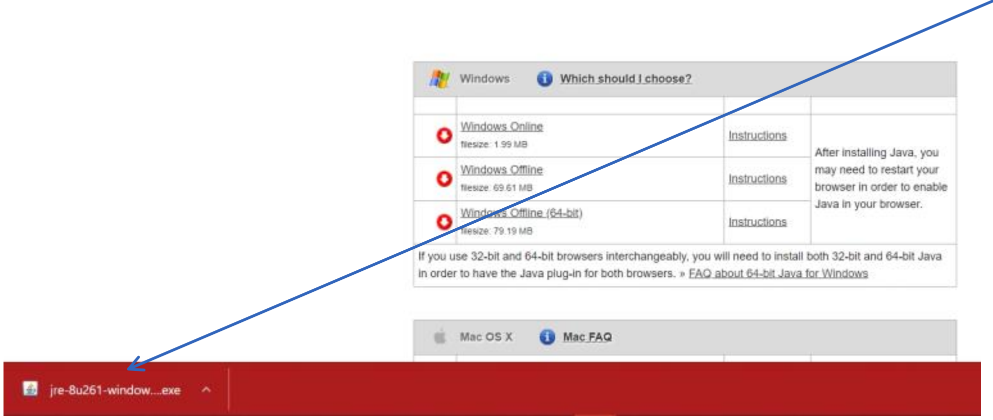
Is this claim valid?

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?**



Windows Which should I choose?

Windows Online filesize: 1.99 MB	Instructions	After installing Java, you may need to restart your browser in order to enable Java in your browser.
Windows Offline filesize: 69.61 MB	Instructions	
Windows Offline (64-bit) filesize: 79.19 MB	Instructions	

If you use 32-bit and 64-bit browsers interchangeably, you will need to install both 32-bit and 64-bit Java in order to have the Java plug-in for both browsers. » FAQ about 64-bit Java for Windows

Mac OS X Mac FAQ

jre-8u261-window....exe

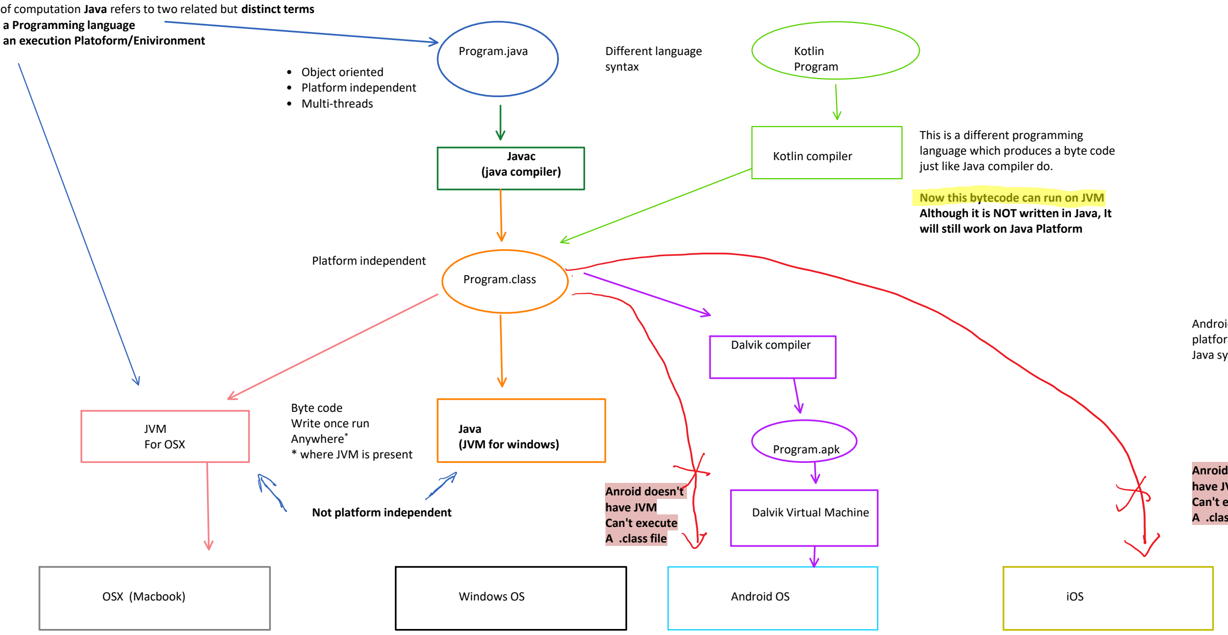
- Why request for a Java download ended up downloading JRE?
- Is JRE and object oriented programming language?
- Is JRE platform independent
- Can you write an object oriented programming language using JRE?

How many Java are there?

Monday, October 19, 2020 12:08 PM

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**



- Object oriented
- Platform independent
- Multi-threads

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

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

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

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

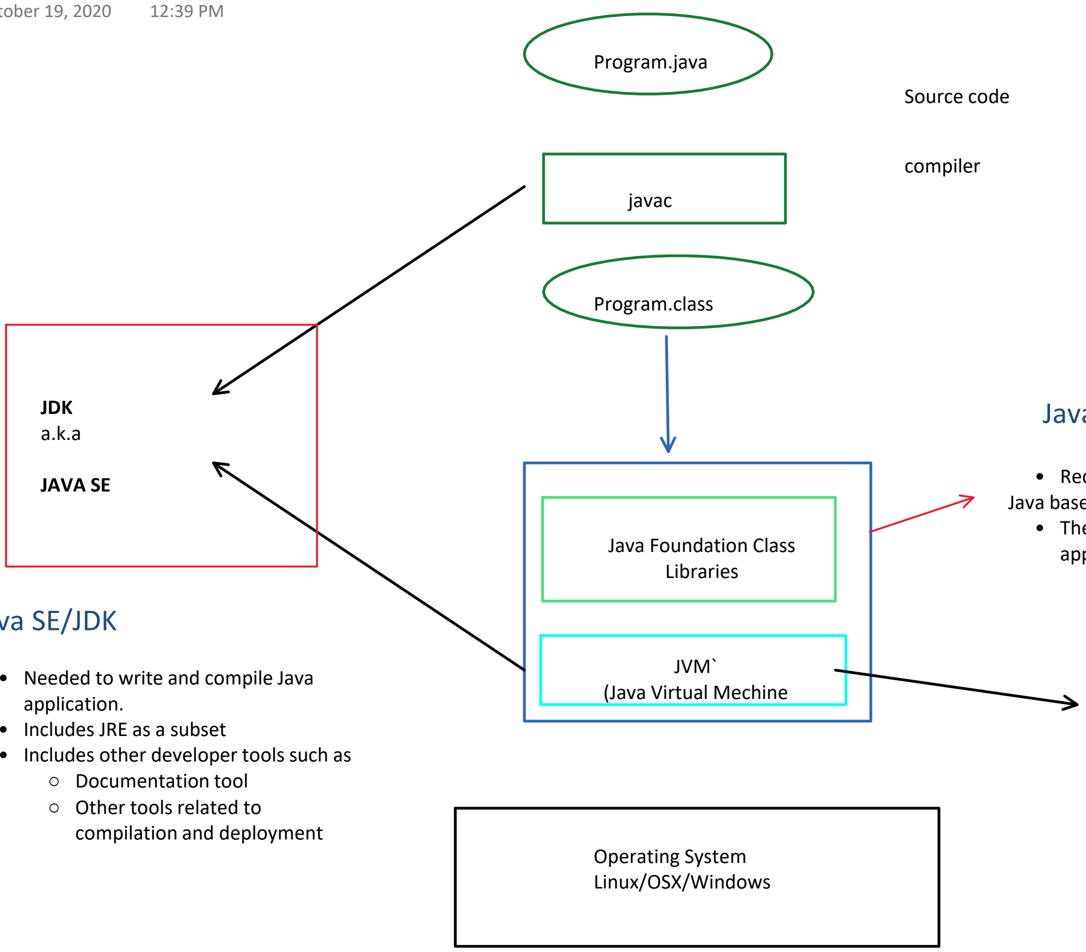
Since the release of Android Studio 3.0 in October 2017, Kotlin is included as an alternative to the standard Java compiler. The Android Kotlin compiler lets the user choose between targeting Java 6 or Java 8 compatible bytecode. Kotlin has been Google's preferred language for Android app development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

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

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

Java Framework

Monday, October 19, 2020 12:39 PM



Program.java

Source code

javac

compiler

Program.class

JDK
a.k.a

JAVA SE

Java Foundation Class Libraries

JVM`
(Java Virtual Mechine

Operating System
Linux/OSX/Windows

Java

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

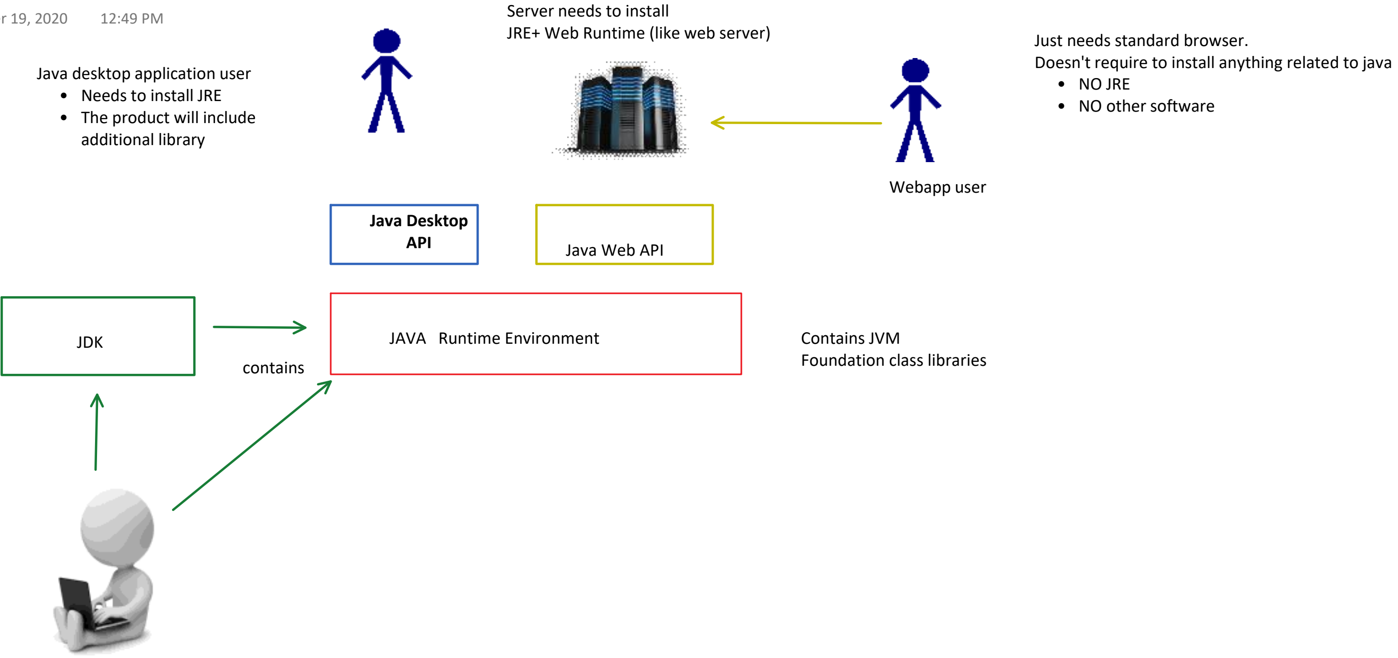
Java SE/JDK

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

Java Environments based on Use cases

Monday, October 19, 2020 12:49 PM



Java desktop application user

- Needs to install JRE
- The product will include additional library

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

Webapp user

Java Desktop API

Java Web API

JDK

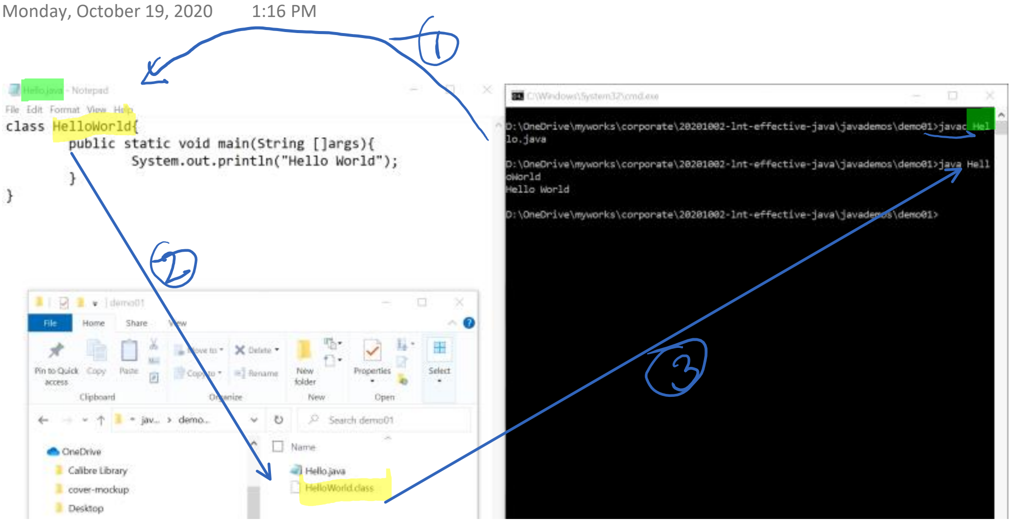
contains

JAVA Runtime Environment

Contains JVM
Foundation class libraries

Compiling and Running Java code

Monday, October 19, 2020 1:16 PM



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

```
D:\OneDrive\myworks\corporate\20201002-lnt-effective-java\javademos\demo01>javac Hel
lo.java

D:\OneDrive\myworks\corporate\20201002-lnt-effective-java\javademos\demo01>java Hell
oWorld
Hello World

D:\OneDrive\myworks\corporate\20201002-lnt-effective-java\javademos\demo01>
```

1. You compile a source file

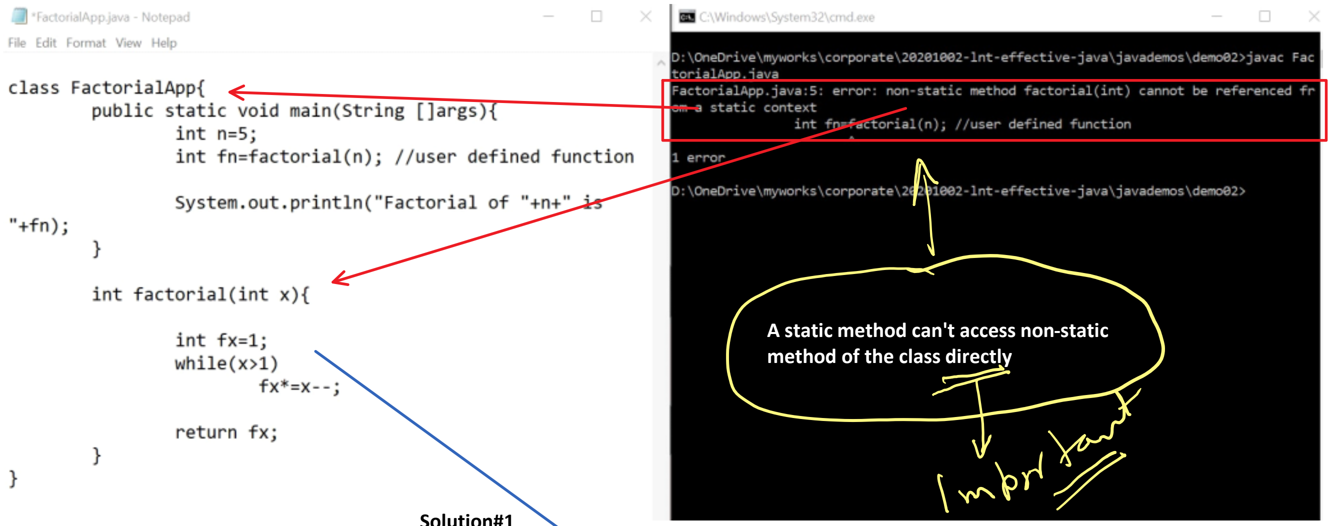
```
c:> javac Hello.java
```

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

Monday, October 19, 2020 1:33 PM



*FactorialApp.java - Notepad
File Edit Format View Help
C:\Windows\System32\cmd.exe

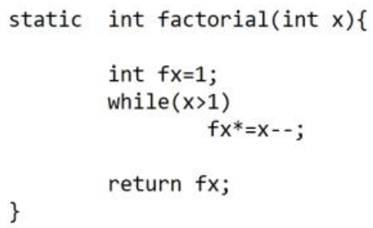
```
class FactorialApp{
	public static void main(String []args){
		int n=5;
		int fn=factorial(n); //user defined function

		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-lnt-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-lnt-effective-java\javademos\demo02>
```

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

Important

Solution#1



```
static  int factorial(int x){

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

	return fx;
}
```

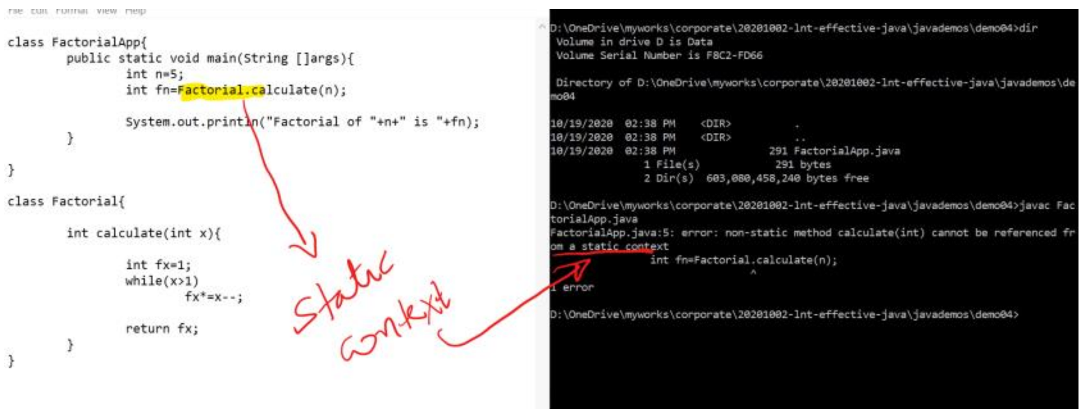
Solution

- There are multiple way to solve this problem
- We will choose different solutions depending on the context

Solution 1

- Make factorial function static

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-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\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-lnt-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-lnt-effective-java\javademos\demo04>
```

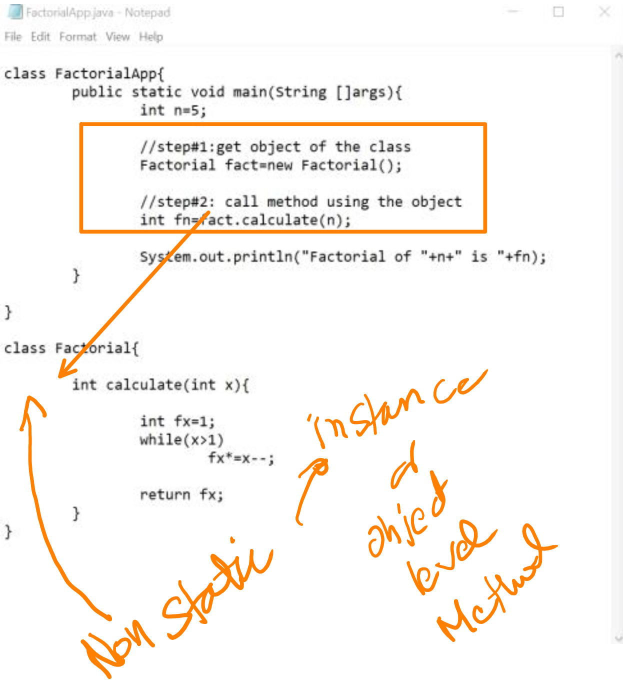
A Class name is a static context. You can call only static methods using Class reference and not non-static methods

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

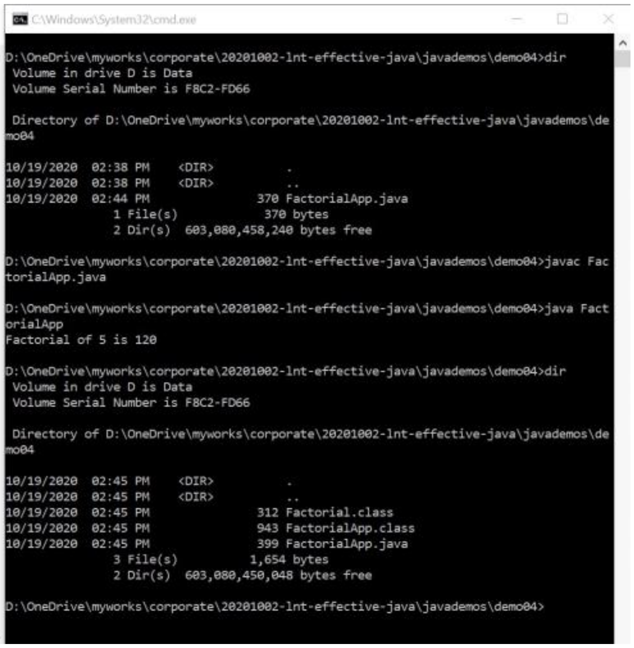
To work with a non-static method, we need an object of the class to refer and use the method.



```java
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

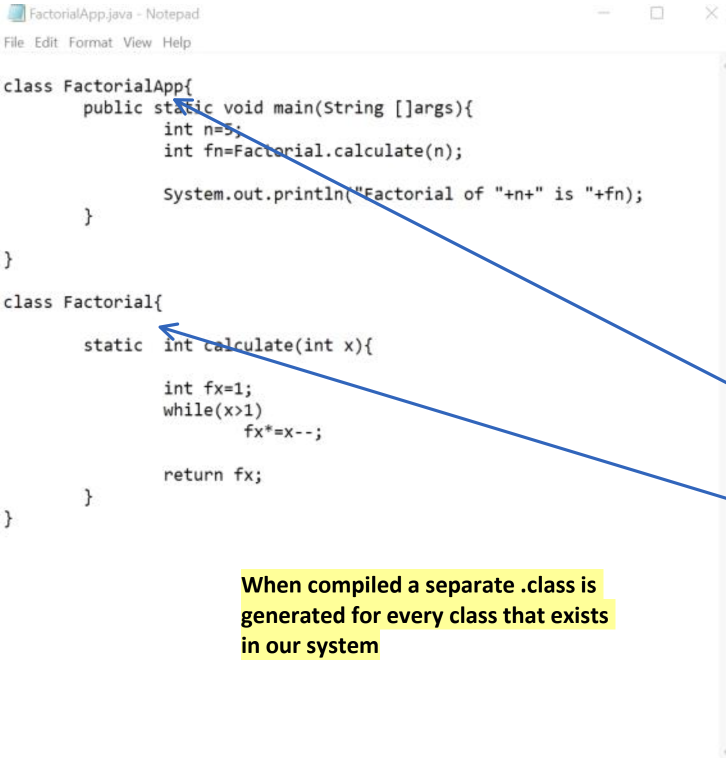
instance
or
object
level
Method



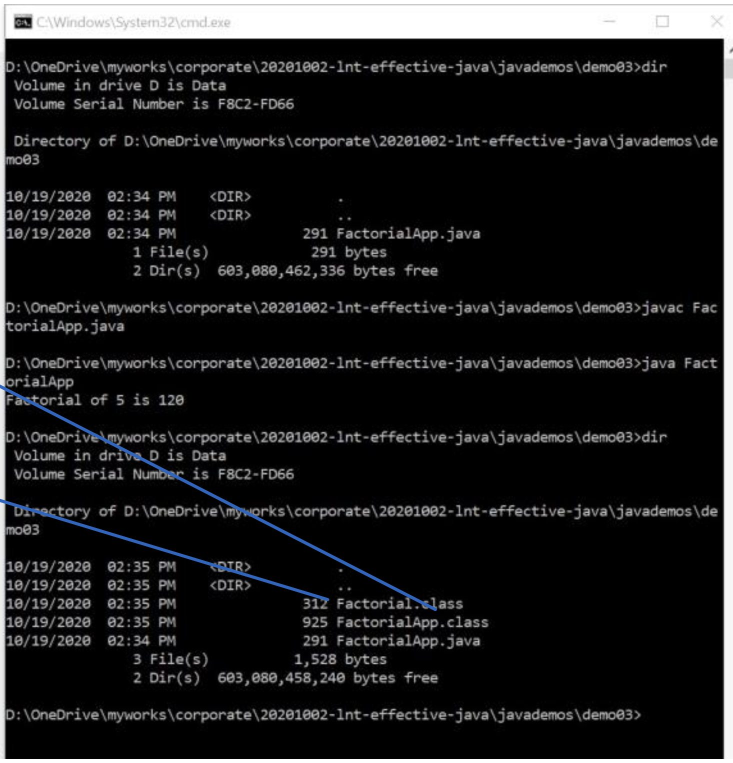
```
D:\OneDrive\myworks\corporate\20201002-lnt-effective-java\javademos\demo04>dir
 Volume in drive D is Data
 Volume Serial Number is F8C2-FD66

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

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 Fac
torialApp.java

D:\OneDrive\myworks\corporate\20201002-lnt-effective-java\javademos\demo04>java Fact
orialApp
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\de
mo04

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



```
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;
        }
}
```

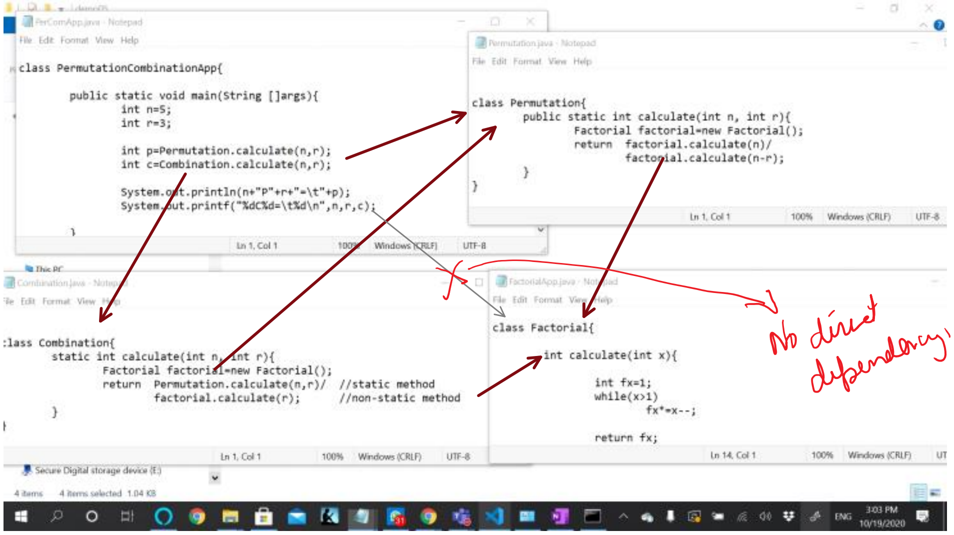


```
D:\OneDrive\myworks\corporate\20201002-lnt-effective-java\javademos\demo03>dir
 Volume in drive D is Data
 Volume Serial Number is F8C2-FD66

 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>
```

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

Class Dependencies

Monday, October 19, 2020 3:03 PM



PerComApp.java - Notepad

```
class PermutationCombinationApp{

        public static void main(String []args){
                int n=5;
                int r=3;

                int p=Permutation.calculate(n,r);
                int c=Combination.calculate(n,r);

                System.out.println(n+"P"+r+"=\t"+p);
                System.out.printf("%dC%d=\t%d\n",n,r,c);

        }
```

Permutation.java - Notepad

```
class Permutation{
        public static int calculate(int n, int r){
                Factorial factorial=new Factorial();
                return  factorial.calculate(n)/
                        factorial.calculate(n-r);

        }
}
```

Combination.java - Notepad

```
class Combination{
        static int calculate(int n, int r){
                Factorial factorial=new Factorial();
                return  Permutation.calculate(n,r)/   //static method
                        factorial.calculate(r);       //non-static method
        }
}
```

FactorialApp.java - Notepad

```
class Factorial{

        int calculate(int x){

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

                return fx;
```

No direct dependency

- 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**

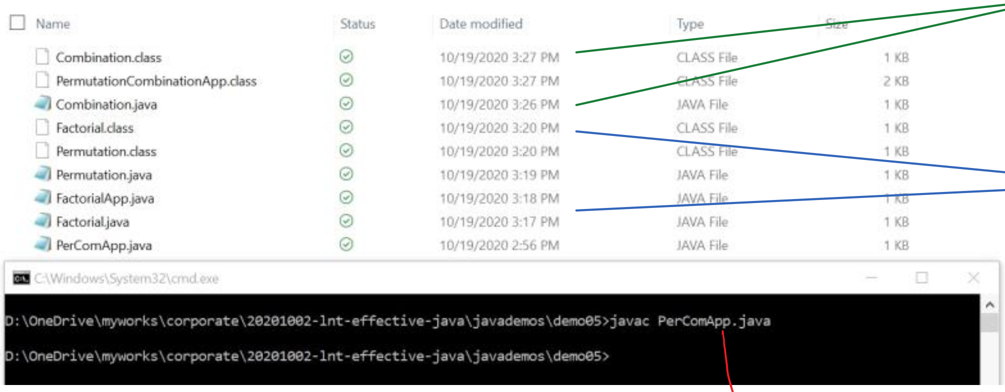
if I have both Combination.class and Combination.java file present, which of them will be used by javac?

- 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

C:\Windows\System32\cmd.exe

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

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

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

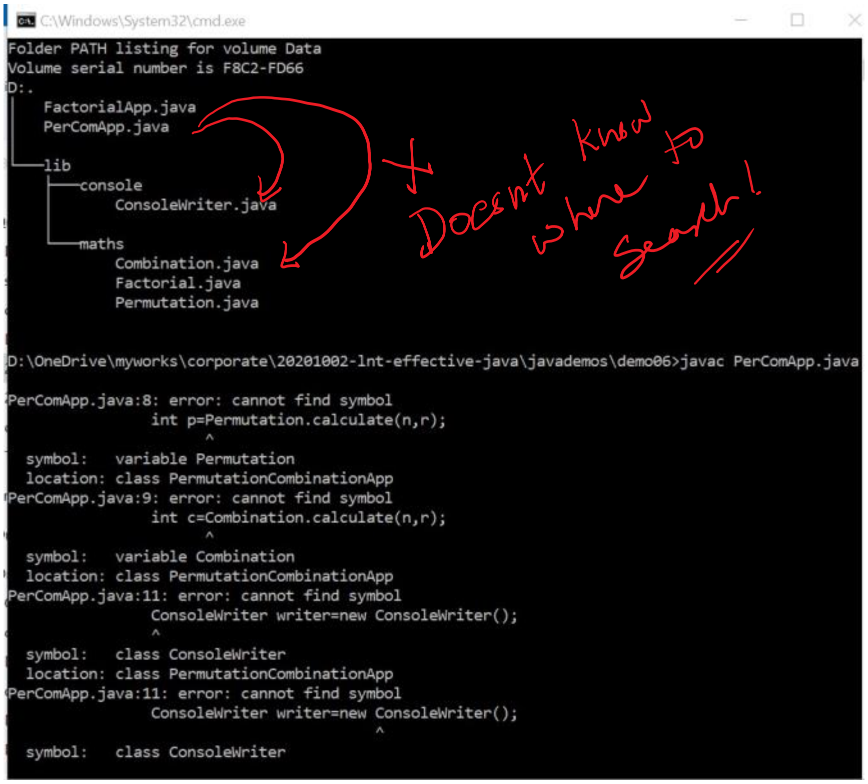
Factorial class has not changed after its
Last modification so it is not recompiled

1

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



```
Folder PATH listing for volume Data
Volume serial number is F8C2-FD66
D:.
│   FactorialApp.java
│   PerComApp.java
│
└───lib
    ├───console
    │       ConsoleWriter.java
    │
    └───maths
            Combination.java
            Factorial.java
            Permutation.java


D:\OneDrive\myworks\corporate\20201002-lnt-effective-java\javademos\demo06>javac PerComApp.java

PerComApp.java:8: error: cannot find symbol
                int p=Permutation.calculate(n,r);
                      ^
  symbol:   variable Permutation
  location: class PermutationCombinationApp
PerComApp.java:9: error: cannot find symbol
                int c=Combination.calculate(n,r);
                      ^
  symbol:   variable Combination
  location: class PermutationCombinationApp
PerComApp.java:11: error: cannot find symbol
                ConsoleWriter writer=new ConsoleWriter();
                ^
  symbol:   class ConsoleWriter
  location: class PermutationCombinationApp
PerComApp.java:11: error: cannot find symbol
                ConsoleWriter writer=new ConsoleWriter();
                                         ^
  symbol:   class ConsoleWriter
```

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 separted using path separator character that varies from one os to another
  - Windows using semicolon
  - Linux/osx uses colon

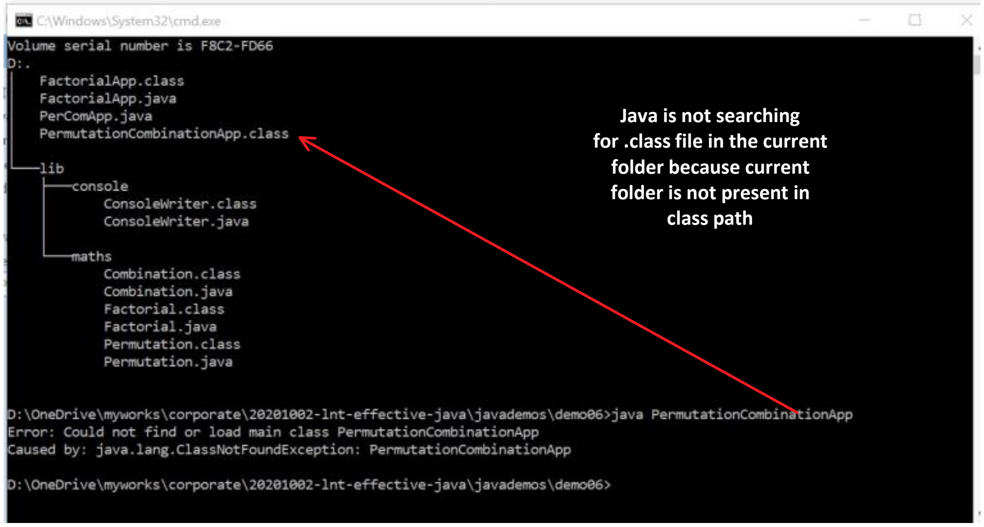
Our class path should look like

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

Current directory

Note

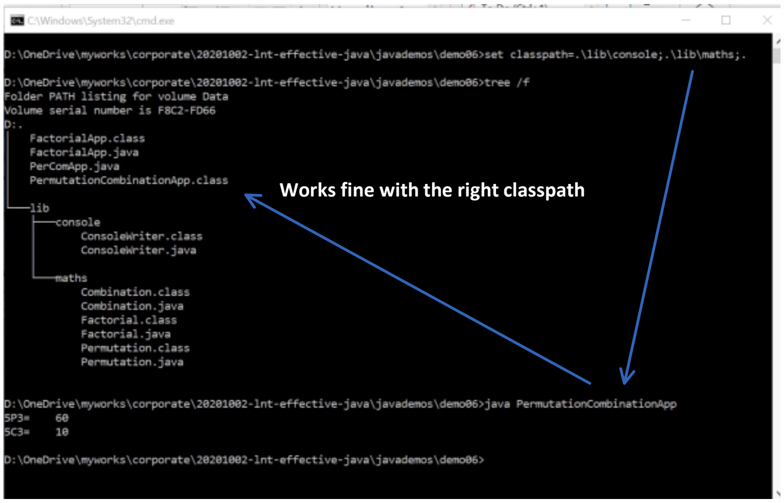
- We have mentioned 3 path here
  - .\lib\console
  - .\lib\maths
  - . (current directory)
- We can't specify just lib
  - We must include right sub directory
- If you are having a classpath, then java/javac doesn't search current directory by default.
  - You must include current directory if you have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



```
Volume serial number is F8C2-FD66
D:.
│   FactorialApp.class
│   FactorialApp.java
│   PerComApp.java
│   PermutationCombinationApp.class
│
└───lib
    ├───console
    │       ConsoleWriter.class
    │       ConsoleWriter.java
    │
    └───maths
            Combination.class
            Combination.java
            Factorial.class
            Factorial.java
            Permutation.class
            Permutation.java


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

Java is not searching for .class file in the current folder because current folder is not present in class path

Works correctly with the Right Path Set

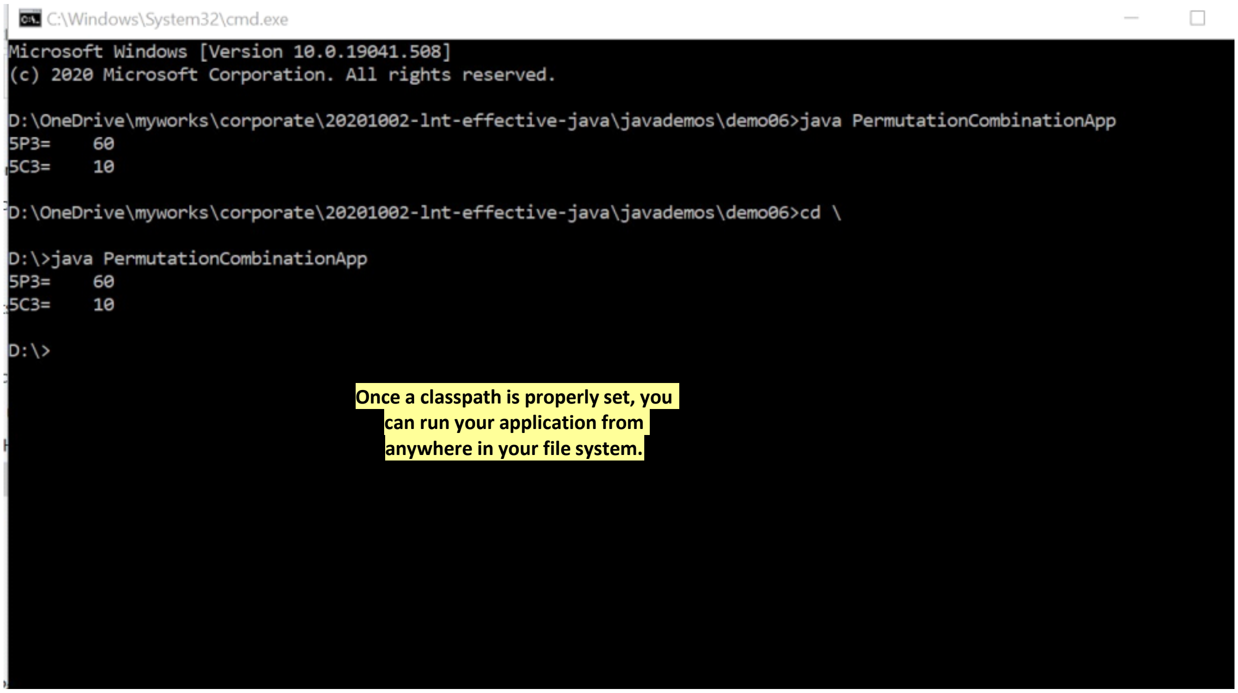


```
D:\OneDrive\myworks\corporate\20201002-lnt-effective-java\javademos\demo06>set classpath=.\lib\console;.\lib\maths;.

D:\OneDrive\myworks\corporate\20201002-lnt-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
│
└───lib
    ├───console
    │       ConsoleWriter.class
    │       ConsoleWriter.java
    │
    └───maths
            Combination.class
            Combination.java
            Factorial.class
            Factorial.java
            Permutation.class
            Permutation.java


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>
```

Works fine with the right classpath

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.**



```
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

Monday, October 19, 2020 4:01 PM

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