Class Notes Link: https://tinyurl.com/COREJAVA7AMNOTES
Mobile No : 7386095600 [Watsapp]
Google Classroom Code: twh7edu

JAVA
—----

1. Introduction
2. Steps to prepare First Java Application
3. Language Fundamentals
4. OOPS **********
5. Wrapper classes
6. Inner Classes
7. Packages
8. String Manipulations
9. Exception Handling
10. Multi Threading
11. IO Streams
12. Networking
13. RMI
14. Internationalization
15. Reflection API
16. Annotations
17. JVM Arch
18. Garbage Collector
19. Regular Expressions
20. Arrays
21. Collection Framework
22. Generics
23. AWT —-- Outdated
24. SWING —--- Outdated
25. Java 8 version Features
26. Java 9 Version Features

—------
—------

35. Java 20 version Features

Steps to Prepare First Application:
—----------------------------------

1. Download And Install Java Software.
2. Download And Install Java Editors.
3. Write Java Program.
4. Save Java File.
5. Compile Java File.
6. Execute Java Program.

Download And Install Java Software:
—---------------------------------

Download the Java softwares from the following links.

JAVA 6:
https://www.oracle.com/in/java/technologies/javase-java-archive-javase6-downloads.html

JAVA 7:
https://www.oracle.com/in/java/technologies/javase/javase7-archive-downloads.html

JAVA 8:
https://www.oracle.com/in/java/technologies/javase/javase8-archive-downloads.html

To install Java Software in our system we have to use the following steps.

1. Double click on the “jdk-6u45-windows-x64.exe” file.
2. Click on the “Next” button.
3. Change the JDK installation location to “C:\Java\jdk1.6.0_45\bin” by clicking on the “Change” button.
4. Click on the “Next” button.
5. Change the JRE installation location to “C:\Java\jre7” by clicking on the “Change” button.
6. Click on the “Next” button.
7. Click on the “Close” button.

Note: Similarly , install JDK1.7 and JDK1.8 versions.

After the installation of the Java Software we have to set the “path” environment variable in order to make available all java commands to the Operating systems[Command prompt].

There are two ways to set the “path” environment variable.

1. Temporary Setup
2. Permanent Setup

Temporary Set:
To set the path environment variable temporarily we have to use the following command in the command prompt.

```
set path=C:\java\jdk1.8.0_202\bin;
```

The above temporary setup is valid up to a single command prompt, it is not valid for all the command prompts, if we want to make available path environment variables for all the command prompts then we have to use Permanent setup.

Permanent Setup:
To set the “path” environment variable permanently we have to use the following steps.

1. Right click on the “This PC”.
2. Select “Properties”.
3. Select the “Advanced System Settings” link.
4. Select the “Advanced” tab[Selected by default]
5. Click on the “Environment Variables…” button.
6. Check whether “path” environment variable exists or not in th user variables part, If the “path” variable exist then select “path” variable , click on “Edit” button and add “C:\java\jdk1.8.0_202\bin” to the “path” environment variable.
7. If the “path” variable does not exist then click on the “New” button and provide the following details.

   Variable Name : path
   Variable Value : C:\java\jdk1.8.0_202\bin

   CLick on the “OK” button.
8. Clicon the “OK” button.

9. Click on the “OK” button.

Open Multiple COmmand prompts and check the Javac option in all the command prompts without setting “path” again and again.

Note: Always Permanent setup of the “path” environment variable is suggestible in our system .

Q)Is it possible to install more than one Java software in a single computer?
—----------
Ans:
—---
Yes, it is possible to install more than one java software in a single computer but we must follow the below conditions.

1. All the Java softwares must be compatible with our present Operating System which we are using in the computer.
2. All the Java softwares must be from different versions.

Q)If we install more than one Java software in a single computer then which java software will come to the command prompt to use?
—----------
Ans:
—---
If we install more than one java software in a single computer then the Java software which we set to the “path” environment variable will come to the command prompt to use.

```
C:\Users\Administrator>set path=C:\java\jdk1.6.0_45\bin;

C:\Users\Administrator>javac -version
javac 1.6.0_45

C:\Users\Administrator>set path=C:\java\jdk1.7.0_80\bin;

C:\Users\Administrator>javac -version
javac 1.7.0_80

C:\Users\Administrator>set path=C:\java\jdk1.8.0_202\bin;
```

```
C:\Users\Administrator>javac -version
javac 1.8.0_202

C:\Users\Administrator>
```

Q) If we install more than one Java software in a single computer and if we set all the Java softwares to the single “path” environment variable then which Java software will come to the command prompt to use?

—----------

Ans:

—---

If we set more than one java software to a single path environment variable then the first java software in the order which we set to the path environment variable will come to the command prompt to use.

```
C:\Users\Administrator>set
path=C:\java\jdk1.6.0_45\bin;C:\java\jdk1.7.0_80\bin;C:\java\jdk1.8.0_
202\bin;

C:\Users\Administrator>javac -version
javac 1.6.0_45

C:\Users\Administrator>set
path=C:\java\jdk1.7.0_80\bin;C:\java\jdk1.8.0_202\bin;C:\java\jdk1.6.0
_45\bin;

C:\Users\Administrator>javac -version
javac 1.7.0_80

C:\Users\Administrator>set
path=C:\java\jdk1.8.0_202\bin;C:\java\jdk1.6.0_45\bin;C:\java\jdk1.7.0
_80\bin;

C:\Users\Administrator>javac -version
javac 1.8.0_202
```

Q)If we set JAVA 1.6 version to the “path” environment variable temporarily and If we set JAVA 1.8 version to the “path” environment variable permanently then which Java version will come to the command prompt to use?

—----------

Ans:

—---

If we set JAVA 1.6 version to the “path” environment variable temporarily and If we set JAVA 1.8 version to the “path” environment variable permanently then the Java version which we set to the “path” environment variable temporarily will come to the command prompt to use, that is JAVA 1.6 version.

```
C:\Users\Administrator>javac -version
javac 1.8.0_202

C:\Users\Administrator>set path=C:\java\jdk1.6.0_45\bin;

C:\Users\Administrator>javac -version
javac 1.6.0_45

C:\Users\Administrator>
```

To switch java from one version to another version easily we have to use batch files.

To use batch files in our system we have to use the following steps.

1. Prepare a batch file with the required set path command for each and every java version.
   a. Take a note pad.
   b. Provide the required set path command.
   c. Save with .bat extension.

   EX:

   java6.bat

   ```
   set path=C:\java\jdk1.6.0_45\bin;
   ```

   java7.bat

   ```
   set path=C:\java\jdk1.7.0_80\bin;
   ```

```
      java8.bat
      set path=C:\java\jdk1.8.0_202\bin;

   2.  Execute batch files as per the requirement.
         a. Open COmmand prompt.
         b. Goto the location where bat files are saved.
         c. Write bat file name and click on enter button.
EX:
C:\Users\Administrator>d:

D:\>cd FullstackJava830

D:\FullstackJava830>cd JAVA7

D:\FullstackJava830\JAVA7>java6.bat

D:\FullstackJava830\JAVA7>set path=C:\java\jdk1.6.0_45\bin;

D:\FullstackJava830\JAVA7>javac -version
javac 1.6.0_45

D:\FullstackJava830\JAVA7>java7.bat

D:\FullstackJava830\JAVA7>set path=C:\java\jdk1.7.0_80\bin;

D:\FullstackJava830\JAVA7>javac -version
javac 1.7.0_80

D:\FullstackJava830\JAVA7>java8.bat

D:\FullstackJava830\JAVA7>set path=C:\java\jdk1.8.0_202\bin;

D:\FullstackJava830\JAVA7>javac -version
javac 1.8.0_202

D:\FullstackJava830\JAVA7>
```

Download And Install Java Editors:
—----------------------------------
Java Editor: Editor is a Software, it will provide a very good environment to write java programs and to save java programs into our system.

EX: Notepad, Editplus, Notepad++, Atom, Sublime text, brackets, VS Code,.....

Note: Editors are suggestible for learning process only, they are not recommended for the real time application development, IDEs are recommended for the real time application development.

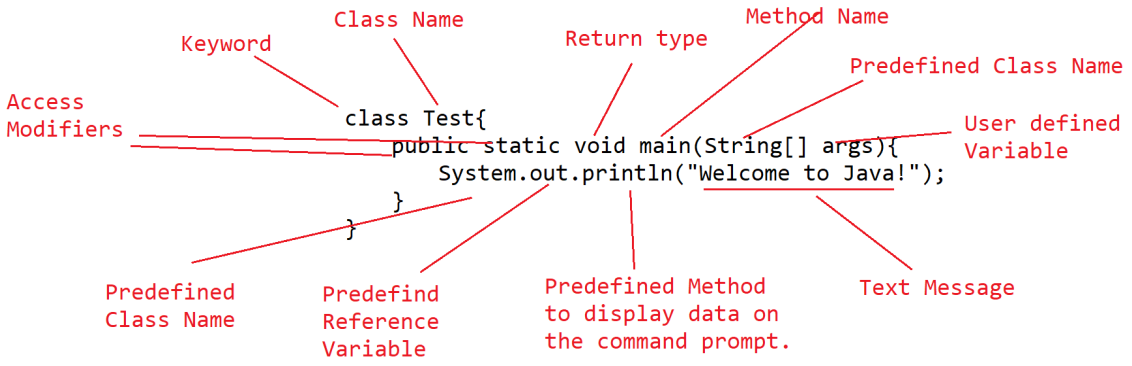
IDE: INtegrated Development Environment.
EX: Eclipse, Intellij Idea, Netbeans,....

Utilization of Editplus:

1. Download and Install Editplus.
    a. Download editplus from the following url.
       https://www.editplus.com/download.html
    b. Double click on the “epp570_4352_64bit.exe” file.
    c. Click on the “Accept” button.
    d. Click on the “Start Copy” button.
    e. Click on the “OK” button.
2. Open Editplus and open Java editor.
    a. Double click on the “Editplus” icon on our desktop.
    b. Click on the “Yes” button.
    c. Click on the “OK” button.
    d. Select the “Trail” button.
    e. Click on the “I Agree” Button.
    f. Select the “File” from the menu.
    g. Select the “New” button from the File.
    h. Select the “Java” from the “New”.

Write Java Program:
—------------------
To Write minimum Java program we need the following elements.

1. Main Class
2. main() method
3. System.out.println() statement to display data on the command prompt.



Save Java File:
—--------------
To Save Java file we have to use the following steps.

1. Select the “File” from Menu.
2. Select the “Save” from the “File”.
   Or
   Ctrl-s
3. Select the location[D:\FullstackJava830\JAVA7].
4. Provide File Name “Test.java”.
5. Click on the SAVE button.

To save a Java File we have to use the following conditions.

1. If the present Java file contains a public element[Class, abstract class, interface, enum] then it is mandatory to save a java file with the public element name only, if we violate this rule then the compiler will raise an error.

2. In the present Java file if no element[Class, abstract class, interface, enum] is public then it is possible to save the present java file with any name like abc.java or xyz.java, in this context JAVA has provided a suggestion to save java file with the main class name instead of using any other name.

EX-1:
File Name: abc.java

```
class Test{
      public static void main(String[] args){
            System.out.println("Welcome to Java!");
      }
}
```

```
D:\FullstackJava830\JAVA7>javac abc.java
D:\FullstackJava830\JAVA7>java Test
Welcome to Java!
```

EX-2:
File Name : Test.java

```
class Test{
      public static void main(String[] args){
            System.out.println("Welcome to Java!");
      }
}
```

```
D:\FullstackJava830\JAVA7>javac Test.java
D:\FullstackJava830\JAVA7>java Test
Welcome to Java!
```

EX-3:
File Name : Test.java

```
public class A{
}
class Test{
      public static void main(String[] args){
            System.out.println("Welcome to Java!");
      }
```

```
}


D:\FullstackJava830\JAVA7>javac Test.java
Test.java:1: error: class A is public, should be declared in a file
named A.java
public class A{
```

EX-4

```
File Name: A.java
public class A{
}
class Test{
      public static void main(String[] args){
            System.out.println("Welcome to Java!");
      }
}


D:\FullstackJava830\JAVA7>javac A.java
D:\FullstackJava830\JAVA7>java Test
Welcome to Java!
```

Q)Is it possible to provide more than one public class in a single Java file?

—----------

Ans:

—---

No, It is not possible to provide more than one public class in a single java file , because if we provide more than one public claSs in a single java file we must save that java file with more than one name, it is not possible in all the Operating systems.

Note: Every Java file must have at most one public class.

EX-5:

```
File Name: A.java
public class A{
}
public class B{
```

```
}
class Test{
      public static void main(String[] args){
            System.out.println("Welcome to Java!");
      }
}
```

```
D:\FullstackJava830\JAVA7>javac A.java
A.java:3: error: class B is public, should be declared in a file named
B.java
public class B{
       ^
```

EX-6:

File Name : B.java

```
public class A{
}
public class B{
}
class Test{
      public static void main(String[] args){
            System.out.println("Welcome to Java!");
      }
}
```

```
D:\FullstackJava830\JAVA7>javac B.java
B.java:1: error: class A is public, should be declared in a file named
A.java
public class A{
       ^
1 error
```

EX-7:

File Name: AB.java

```
public class A{
}
public class B{
}
class Test{
```

```
      public static void main(String[] args){
            System.out.println("Welcome to Java!");
      }
}
```

```
D:\FullstackJava830\JAVA7>javac AB.java
AB.java:1: error: class A is public, should be declared in a file
named A.java
public class A{
       ^
AB.java:3: error: class B is public, should be declared in a file
named B.java
public class B{
       ^
```

Compile Java File:

—------------------

Q)What is the requirement to perform Compilation over a Java file?

—---

Ans:

—----

The main purpose of the compilation is

1. To translate the program from High level language[Developers understandable Language] to the low level language[JVM Understandable Language].
2. To Check the developers mistakes in the program.

To perform the Compilation, JAVA has provided a predefined command in the form of “javac”.

Syntax:

```
javac fileName.java
```

EX:

```
D:\FullstackJava830\JAVA7>javac Test.java
```

When we provide the above command on the command prompt then the Operating System will perform the following actions.

1. The Operating System will take the provided javac command from the command prompt.
2. The Operating System will search for the javac command in its internal commands list and at the locations referred to the “path” environment variable.
3. If the “javac” is not available at both the locations then the Operating System will provide the following message.

   'javac' is not recognized as an internal or external command, operable program or batch file.

4. To make available all java commands to the operating system we have to set the “path” environment variable to the location where all java commands exist, that is C:\Java\jdk1.8.0\bin.

   set path=C:\Java\jdk1.8.0\bin;

5. If the “javac” command is identified by the Operating SYstem through the “path” environment variable then the Operating System will execute the “javac” command.
6. When the “javac” command is executed by the Operating System, automatically compiler software will be activated by the Operating system, with this the java compiler will perform the following actions.

   a. Compiler will take the provided java file name from the command prompt and search for it in the current location.
   b. If the provided java file does not exist then the compiler will provide the following message on the command prompt.
      javac: file not found: Test.java
   c. If the provided file is identified at the current location then the compiler will start compilation right from the starting point of the file to the ending point of the file.
   d. In the compilation process, if the compiler identifies any error then the compiler will provide the respective error message on the command prompt.
   e. In the compilation process, if no error is identified in the java file then the compiler will finish its compilation and it will generate .class files.

Note: In Java, generating .class files is not at all dependent on the number of java files which we compiled, it is totally dependent on the number of classes, abstract classes, interfaces, enums and inner classes which we provided in the present java file.

EX-1:
File Name: Test.java

```
class Test{
      public static void main(String[] args){
            System.out.println("First Java Application");
      }
}
```

```
D:\FullstackJava830\JAVA7>javac Test.java

D:\FullstackJava830\JAVA7>dir
 Volume in drive D is Data
 Volume Serial Number is 68E9-29EB

 Directory of D:\FullstackJava830\JAVA7

10-11-2023  07:56    <DIR>          .
10-11-2023  07:56    <DIR>          ..
10-11-2023  07:43            43,661 Compilation.png
08-11-2023  07:34                35 java6.bat
08-11-2023  07:35                35 java7.bat
08-11-2023  07:36                36 java8.bat
10-11-2023  07:56               424 Test.class
10-11-2023  07:56               111 Test.java
               6 File(s)         44,302 bytes
               2 Dir(s)  350,247,858,176 bytes free

D:\FullstackJava830\JAVA7>
```

EX-2:
File Name : Test.java

```
enum E{
}
interface I{
}
abstract class A{
}
class B{
     class C    {
     }
}
class Test{
     public static void main(String[] args){
          System.out.println("First Java Application");
     }
}
```

```
D:\FullstackJava830\JAVA7>javac Test.java

D:\FullstackJava830\JAVA7>dir
 Volume in drive D is Data
 Volume Serial Number is 68E9-29EB

 Directory of D:\FullstackJava830\JAVA7

10-11-2023  08:00    <DIR>          .
10-11-2023  08:00    <DIR>          ..
10-11-2023  08:00               179 A.class
10-11-2023  08:00               270 B$C.class
10-11-2023  08:00               223 B.class
10-11-2023  07:43            43,661 Compilation.png
10-11-2023  08:00               611 E.class
10-11-2023  08:00                86 I.class
08-11-2023  07:34                35 java6.bat
08-11-2023  07:35                35 java7.bat
08-11-2023  07:36                36 java8.bat
10-11-2023  08:00               424 Test.class
10-11-2023  07:59               191 Test.java
```

```
10-11-2023  07:56               111 Test.java.bak
              12 File(s)         45,862 bytes
               2 Dir(s)  350,247,858,176 bytes free

D:\FullstackJava830\JAVA7>
```

If we want to compile the java file from the current location and if we want to send the generated .class files to some other location then we have to use -d option along with the javac command.

```
javac -d TargetLocation FileName.java
```

EX:

File Name: D:\FullstackJava830\JAVA7\ Test.java

```
enum E{
}
interface I{
}
abstract class A{
}
class B{
      class C    {
      }
}
class Test{
      public static void main(String[] args){
            System.out.println("First Java Application");
      }
}
```

```
D:\FullstackJava830\JAVA7>javac -d E:\abc Test.java

D:\FullstackJava830\JAVA7>dir
 Volume in drive D is Data
 Volume Serial Number is 68E9-29EB

 Directory of D:\FullstackJava830\JAVA7

10-11-2023  08:05    <DIR>          .
```

```
10-11-2023  08:05    <DIR>          ..
10-11-2023  07:43            43,661 Compilation.png
08-11-2023  07:34                35 java6.bat
08-11-2023  07:35                35 java7.bat
08-11-2023  07:36                36 java8.bat
10-11-2023  07:59               191 Test.java
10-11-2023  07:56               111 Test.java.bak
               6 File(s)         44,069 bytes
               2 Dir(s)  350,247,858,176 bytes free

D:\FullstackJava830\JAVA7>e:

E:\>cd abc

E:\abc>dir
 Volume in drive E is New Volume
 Volume Serial Number is 7E96-6D63

 Directory of E:\abc

10-11-2023  08:06    <DIR>          .
10-11-2023  08:06    <DIR>          ..
10-11-2023  08:06               179 A.class
10-11-2023  08:06               270 B$C.class
10-11-2023  08:06               223 B.class
10-11-2023  08:06               611 E.class
10-11-2023  08:06                86 I.class
10-11-2023  08:06               424 Test.class
               6 File(s)          1,793 bytes
               2 Dir(s)  274,683,916,288 bytes free

E:\abc>
```

If we use a package in the present java file then we have to use -d option along with javac command in order to create folder structure with respect to the package name.

EX:
File Name: D:\FullstackJava830\JAVA7\ Test.java

```
package com.durgasoft.core;
```

```
enum E{
}
interface I{
}
abstract class A{
}
class B{
     class C    {
     }
}
class Test{
     public static void main(String[] args){
          System.out.println("First Java Application");
     }
}


D:\FullstackJava830\JAVA7>javac -d E:\abc Test.java

D:\FullstackJava830\JAVA7>e:

E:\abc>dir
 Volume in drive E is New Volume
 Volume Serial Number is 7E96-6D63

 Directory of E:\abc

10-11-2023  08:14    <DIR>          .
10-11-2023  08:14    <DIR>          ..
10-11-2023  08:14    <DIR>          com
               0 File(s)              0 bytes
               3 Dir(s)  274,683,916,288 bytes free

E:\abc>cd com

E:\abc\com>dir
 Volume in drive E is New Volume
 Volume Serial Number is 7E96-6D63

 Directory of E:\abc\com
```

```
10-11-2023  08:14    <DIR>          .
10-11-2023  08:14    <DIR>          ..
10-11-2023  08:14    <DIR>          durgasoft
               0 File(s)              0 bytes
               3 Dir(s)  274,683,916,288 bytes free

E:\abc\com>cd durgasoft

E:\abc\com\durgasoft>dir
 Volume in drive E is New Volume
 Volume Serial Number is 7E96-6D63

 Directory of E:\abc\com\durgasoft

10-11-2023  08:14    <DIR>          .
10-11-2023  08:14    <DIR>          ..
10-11-2023  08:14    <DIR>          core
               0 File(s)              0 bytes
               3 Dir(s)  274,683,916,288 bytes free

E:\abc\com\durgasoft>cd core

E:\abc\com\durgasoft\core>dir
 Volume in drive E is New Volume
 Volume Serial Number is 7E96-6D63

 Directory of E:\abc\com\durgasoft\core

10-11-2023  08:14    <DIR>          .
10-11-2023  08:14    <DIR>          ..
10-11-2023  08:14               198 A.class
10-11-2023  08:14               346 B$C.class
10-11-2023  08:14               261 B.class
10-11-2023  08:14               706 E.class
10-11-2023  08:14               105 I.class
10-11-2023  08:14               443 Test.class
               6 File(s)          2,059 bytes
               2 Dir(s)  274,683,916,288 bytes free
```

E:\abc\com\durgasoft\core>

Q)Is it possible to compile more than one java file by using a single javac command?
—--
Ans:
—---
Yes, it is possible to compile more than one java file with a single javac command with the following cases.

Case#1:
If all the java files are interdependent on each other and if we compile the first java file , automatically the compiler will compile all the dependent java files.
EX:
Test.java
```
class Test{
      A a = new A();
}
```

A.java
```
class A{
      B b = new B();
}
```

B.java
```
class B{
      C c = new C();
}
```

C.java
```
class C{

}
```

D:\FullstackJava830\JAVA7>javac Test.java

D:\FullstackJava830\JAVA7>dir
 Volume in drive D is Data

```
 Volume Serial Number is 68E9-29EB

 Directory of D:\FullstackJava830\JAVA7

11-11-2023  07:46    <DIR>          .
11-11-2023  07:46    <DIR>          ..
11-11-2023  07:46               231 A.class
11-11-2023  07:44                30 A.java
11-11-2023  07:46               231 B.class
11-11-2023  07:44                30 B.java
11-11-2023  07:46               176 C.class
11-11-2023  07:44                16 C.java
10-11-2023  10:46            45,365 Compilation.png
08-11-2023  07:34                35 java6.bat
08-11-2023  07:35                35 java7.bat
08-11-2023  07:36                36 java8.bat
11-11-2023  07:46               237 Test.class
11-11-2023  07:45                33 Test.java
10-11-2023  08:12               220 Test.java.bak
              13 File(s)         46,675 bytes
               2 Dir(s)  350,244,384,768 bytes free

D:\FullstackJava830\JAVA7>
```

Case#2:
By Providing Multiple Java files along with a single javac command with the space separator.

EX:
Test.java

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

A.java

```
class A{

}
```

B.java

```
class B{
}
```

C.java

```
class C{
}
```

```
D:\FullstackJava830\JAVA7>javac Test.java A.java B.java C.java

D:\FullstackJava830\JAVA7>dir
 Volume in drive D is Data
 Volume Serial Number is 68E9-29EB

 Directory of D:\FullstackJava830\JAVA7

11-11-2023  07:50    <DIR>          .
11-11-2023  07:50    <DIR>          ..
11-11-2023  07:50               176 A.class
11-11-2023  07:49                16 A.java
11-11-2023  07:50               176 B.class
11-11-2023  07:49                16 B.java
11-11-2023  07:50               176 C.class
11-11-2023  07:49                16 C.java
10-11-2023  10:46            45,365 Compilation.png
08-11-2023  07:34                35 java6.bat
08-11-2023  07:35                35 java7.bat
08-11-2023  07:36                36 java8.bat
11-11-2023  07:50               253 Test.class
11-11-2023  07:49                62 Test.java
              12 File(s)         46,362 bytes
               2 Dir(s)  350,244,384,768 bytes free

D:\FullstackJava830\JAVA7>
```

Case#3:
By Using * notation.

To compile all java files which are available at the current location then we have to use * notation.

```
D:\FullstackJava830\JAVA7>javac *.java
```

EX:

Test.java

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

A.java

```
class A{

}
```

B.java

```
class B{
}
```

C.java

```
class C{
}
```

```
:\FullstackJava830\JAVA7>javac *.java

D:\FullstackJava830\JAVA7>dir
 Volume in drive D is Data
 Volume Serial Number is 68E9-29EB

 Directory of D:\FullstackJava830\JAVA7

11-11-2023  07:53    <DIR>          .
11-11-2023  07:53    <DIR>          ..
11-11-2023  07:53               176 A.class
11-11-2023  07:49                16 A.java
11-11-2023  07:53               176 B.class
```

```
11-11-2023  07:49                16 B.java
11-11-2023  07:53               176 C.class
11-11-2023  07:49                16 C.java
10-11-2023  10:46            45,365 Compilation.png
08-11-2023  07:34                35 java6.bat
08-11-2023  07:35                35 java7.bat
08-11-2023  07:36                36 java8.bat
11-11-2023  07:53               253 Test.class
11-11-2023  07:49                62 Test.java
              12 File(s)         46,362 bytes
               2 Dir(s)  350,244,384,768 bytes free

D:\FullstackJava830\JAVA7>
```

To compile all java files which are having a common prefix and it may end with anything then we have to use the following command.

```
javac prefixName*.java
EX:
D:\FullstackJava830\JAVA7>javac Employee*.java
```

To compile all the java files which are having a common suffix and which may start with anything then we have to use the following command.

```
javac *SuffixName.java
EX:
D:\FullstackJava830\JAVA7>javac *Address.java
```

To Compile all the java files which contain a common word then we have to use the following command.

```
javac *CommonWord*.java
EX:
D:\FullstackJava830\JAVA7>javac *Account*.java
```

Execute Java Program.
—----------------------
To execute a Java program we have to use the following command on the command prompt from the location where the main class's .class file exist.

```
java MainClassName
```

EX:

```
D:\FullstackJava830\JAVA7>java Test
```

When we provide the above command on the command prompt , the Operating System will perform the following actions.

1. The Operating System will take the "java" command from the command prompt.
2. The Operating System will search for the java command in its internal commands list and at the locations provided by the "path" environment variable.
3. When the operating system identifies the "java" command through the "path" environment variable provided locations , The operating system will execute java command, with this the JVM software will be activated and the JVM software will perform the following actions.

    a. JVM will take the main class name from the command prompt and the JVM will search for the main class's .class file at the current location, at the Java predefined library and at the locations provided by the "classpath" environment variable.
    b. If the required main class's .class file does not exist at all the above locations then the JVM will provide the following messages.

    ```
    JAVA6: java.lang.NoClassDefFoundError: Test
    JAVA7: Error: could not find or load main class Test
    ```

    c. If the required main class's .class file is available at the different location then we have to set "classpath"

environment variable to that location in order to provide that location details to the JVM.

D:\FullstackJava830\JAVA7>set classpath=E:/abc;

d. If the required main class's .class file exists at either of the above locations then the JVM will load the main class bytecode to the memory.
e. After loading the main class bytecode to the memory, JVM will search for the main() method in the loaded class.
f. If the required main() method does not exist in the loaded main class then the JVM will provide the following exception messages.

JAVA6: java.lang.NoSuchMethodError: main
JAVA7: Error: main method not found in class Test, please
define the main method as:
public static void main(String[] args)

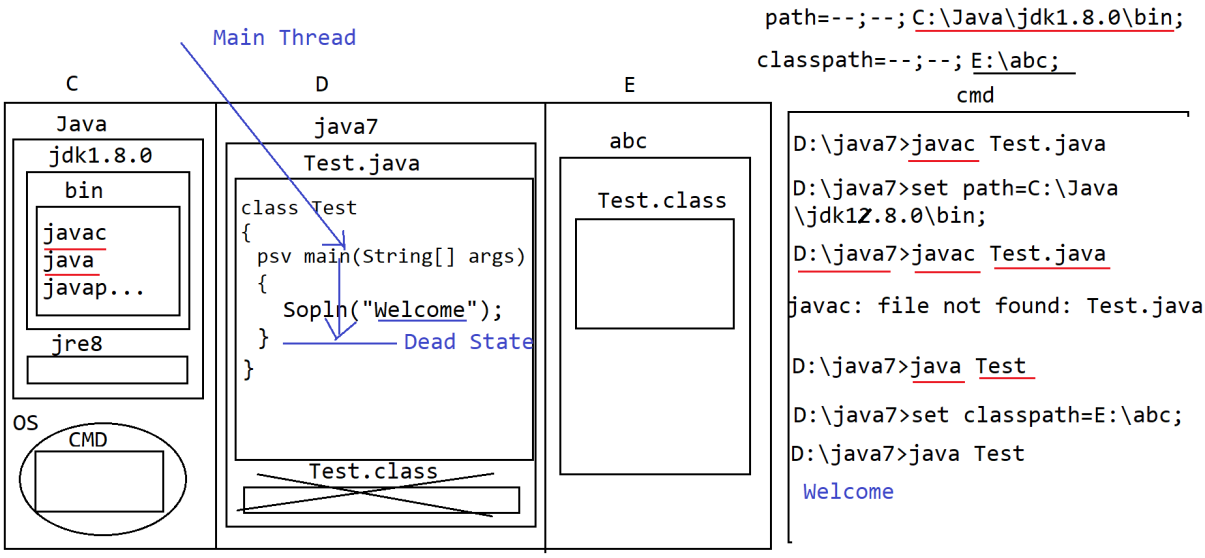
g. If the main() method exists in the loaded main class then the JVM will create a thread called Main Thread to execute the main() method.
h. When JVM comes to the ending point of the main() method, JVM will keep the Main Thread in the Dead State, with this JVM will stop all of its internal processing and JVM will goto shutdown mode.

Q)What is the difference between "path" environment variable and "classpath" environment variable?
—----------
Ans:
—---
"path" environment variable is able to provide the location details of all java commands to the operating system.

set path=C:\Java\jdk1.8.0\bin;

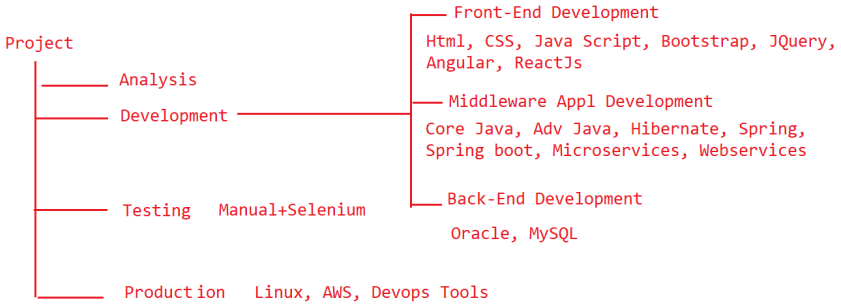
"classpath" environment variable is able to provide the location details of the .class files to the compiler and JVM.

set classpath=E:\abc;



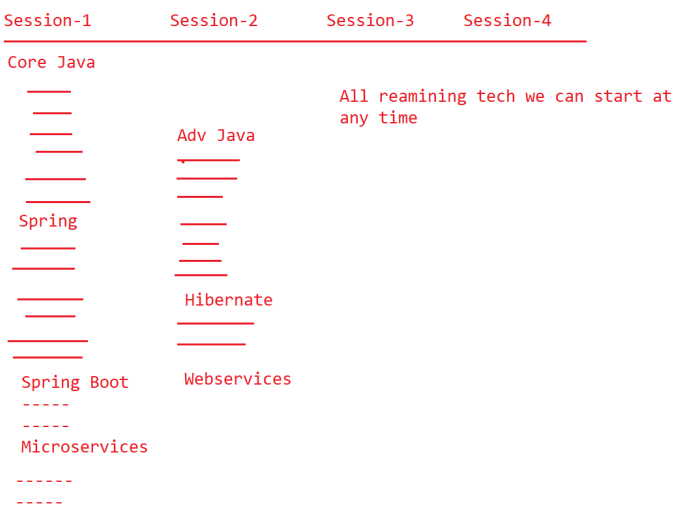
Fullstack:

-- Core Java
-- Adv Java
-- Hibernate
-- Spring
-- Spring Boot
-- Microservices
-- Webservices
-- Oracle
-- Basic UI
-- Angular /React JS
-- Linux
-- AWS
-- Devops
-- Testing Tools



Fullstack:

-- Core Java
-- Adv Java
-- Hibernate
-- Spring
-- Spring Boot
-- Microservices
-- Webservices
-- Oracle
-- Basic UI
-- Angular /React JS
-- Linux
-- AWS
-- Devops
-- Testing Tools



Advantages of this Course:
—----------

1. All My classes Live videos records will be provided for more than 1 year.
2. All my classes material in the form of Hard copy[chargeable] for Offline and softcopy for online and Offline students.
3. Free access to the class room notes.
4. Free access to the classroom applications.
5. All applications live Application Development.
6. Course completion certificate will be provided at the end of the course.

7. Company drives will be provided for All the Students
8. Concepts wise Written tests.
9. Mock interviews[Optional]
10. Working with Multiple databases
    Oracle, MySQL,....
11. Working with Multiple Servers
    a. Weblogic
    b. Wildfly
    c. Glashfish
    d. Tomcat
12. Working with Multiple IDEs
    a. Eclipse
    b. IntelliJ Idea
    c. Netbeans
13. Interview Questions.
14. Up to Latest version[Up to Latest JAVA 20]
15. Classes are mandatory from Monday to Saturday, depending on the situation we will conduct the classes even on Sundays also.

JAVA:
—-----
7:0AM to 8:30AM
8:30Am to 10:00 AM
10:00AM to 11:30AM

Batch#1
Core Java

3 months

Batch#2
Basic Java Package

1. Core Java
2. Adv Java
3. Oracle
4. Basic UI

4 to 5

Batch#3
Complete Java Package

1. Core Java
2. Adv Java
3. Hibernate
4. Spring
5. Spring Boot
6. Microservices
7. Webservices
8. Oracle
9. Basic UI

5 to 6

Batch#4
Full stacl Package

1. Core Java
2. Adv Java
3. Hibernate
4. Spring
5. Spring Boot
6. Microservices
7. Webservices
8. Oracle
9. Basic UI
10. Angular/React JS
11. Linux
12. AWS
13. Devops
14. Testing Tools

7 to 8

Offline : 8852 8852 63/64/96/97

Online: 7386095600,8885252627, 9246212143,
Online mail: durgasoftonlinetraining@gmail.com
Website: www.durgasoft.com