⑤ First Java program and Main Method in JAVA

26/10/22

→ In Java 9 → jshell feature
↳ we can write one line of code and execute.

→ jshell > System.out.print (6)

① ⇒ 6+3 =
⇒ $1 ==> 9

② 5+72=77
$2 ==> 77

③ $1 + $2 = 9+77
$3 ⇒ 86

→ jshell gives some commands to follow
↳ jshell > / + Enter
↳ shows all commands in jshell

Ex. jshell > /vars
int $1 = 5
$2 = 72

jshell > /save Code.java.text → Saving code
jshell > /reset → resetting state
jshell > /vars → Variables got Reset

→ jshell was introduced in Java 9
↳ Intention to learn Java Easily.
↳ testing small code
↳ not for building Applications.

① Editor :- for typing Code

② IDE :- Integrated Development Environment
↳ Type, Compile, Run, De-bug, Test

Ex IDE's :- Eclipse, Intellij idea.

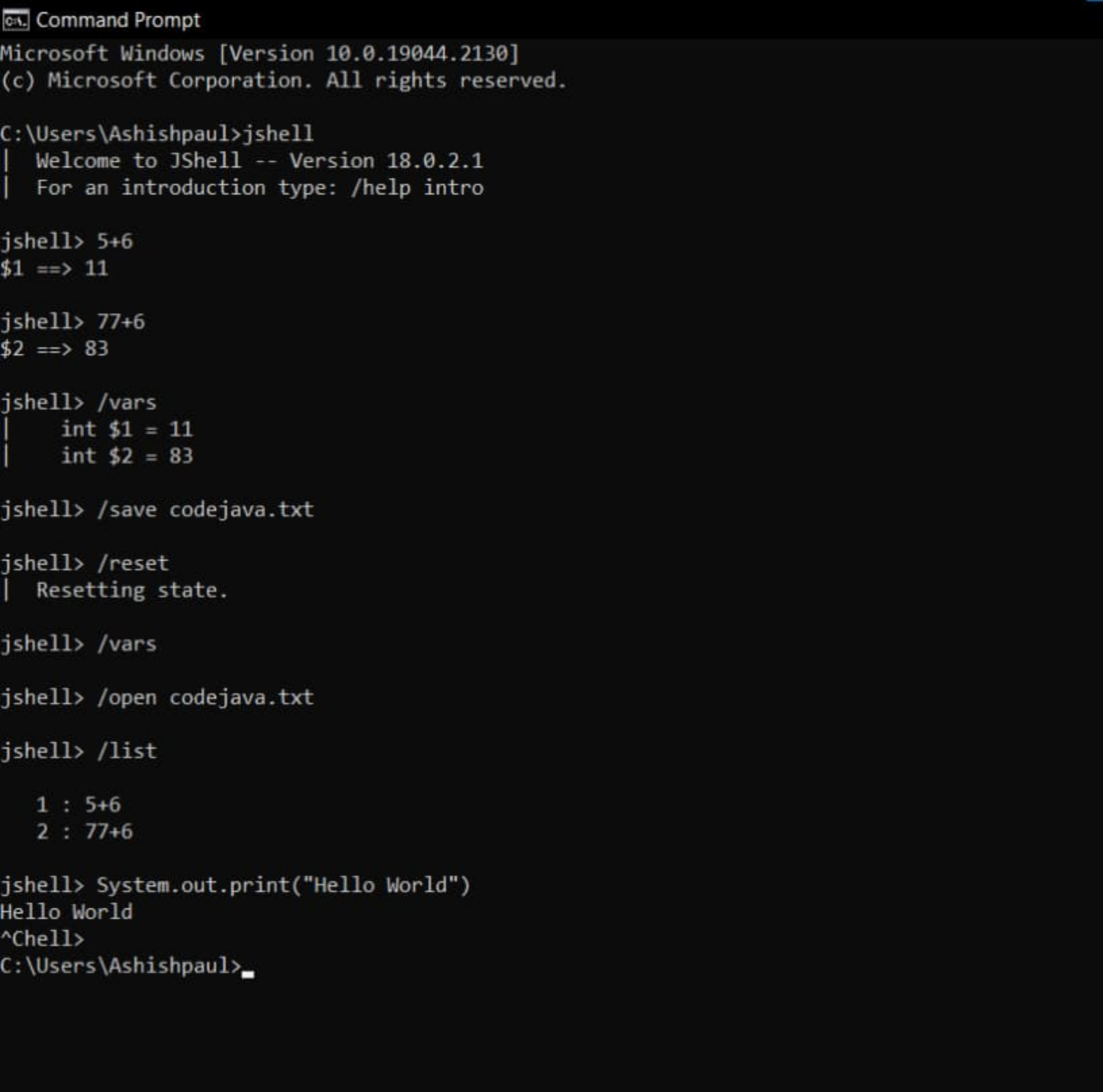
→ Java is called Verbose Code.
↳ we have to Explain Structure & Code
↳ Mention Everything.

→ Every statement need Semi Colon after it. (";")

① Starting Point :-

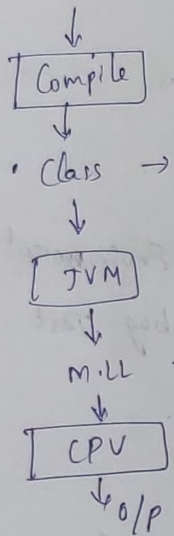
```
Public static Void main (String a[]) {

}
```

→ Everthing in Java should be a part of a class.


Command Prompt
Microsoft Windows [Version 10.0.19044.2130]
(c) Microsoft Corporation. All rights reserved.
C:\Users\Ashishpaul>jshell
| Welcome to JShell -- Version 18.0.2.1
| For an introduction type: /help intro
jshell> 5+6
$1 ==> 11
jshell> 77+6
$2 ==> 83
jshell> /vars
| int $1 = 11
| int $2 = 83
jshell> /save codejava.txt
jshell> /reset
| Resetting state.
jshell> /vars
jshell> /open codejava.txt
jshell> /list
1 : 5+6
2 : 77+6
jshell> System.out.print("Hello World")
Hello World
^Chell>
C:\Users\Ashishpaul>

① Compiling and Interpreting Java Code

① .Java → (Source Code)



(i) Compile :-

→ for Compiling we have to use Javac Command.

(ii) Interpret :-

→ using Java Command we run Code

Eg: Java firstCode.Java

Javac <file name>

Javac FirsCode.Java

↓ Compile

Java firstCode.class.

↓ Interpret

Java FirstCode

↓

output.

→ ~~we directly Interpret~~

→ In upcomming Version of Java Code Can Interpreted directly. it is allowed. In java new versions

→ Note recommended.

→ Recommand to Compile and Interpret Java Code

Eg: Javac firstCode.Java → Compiling.

Java firstCode → Interpret.

→ we have to run Code with Class name.

Eg: Javac <file name>

Java <Class name>

→ File name

Code :- firstCode.Java

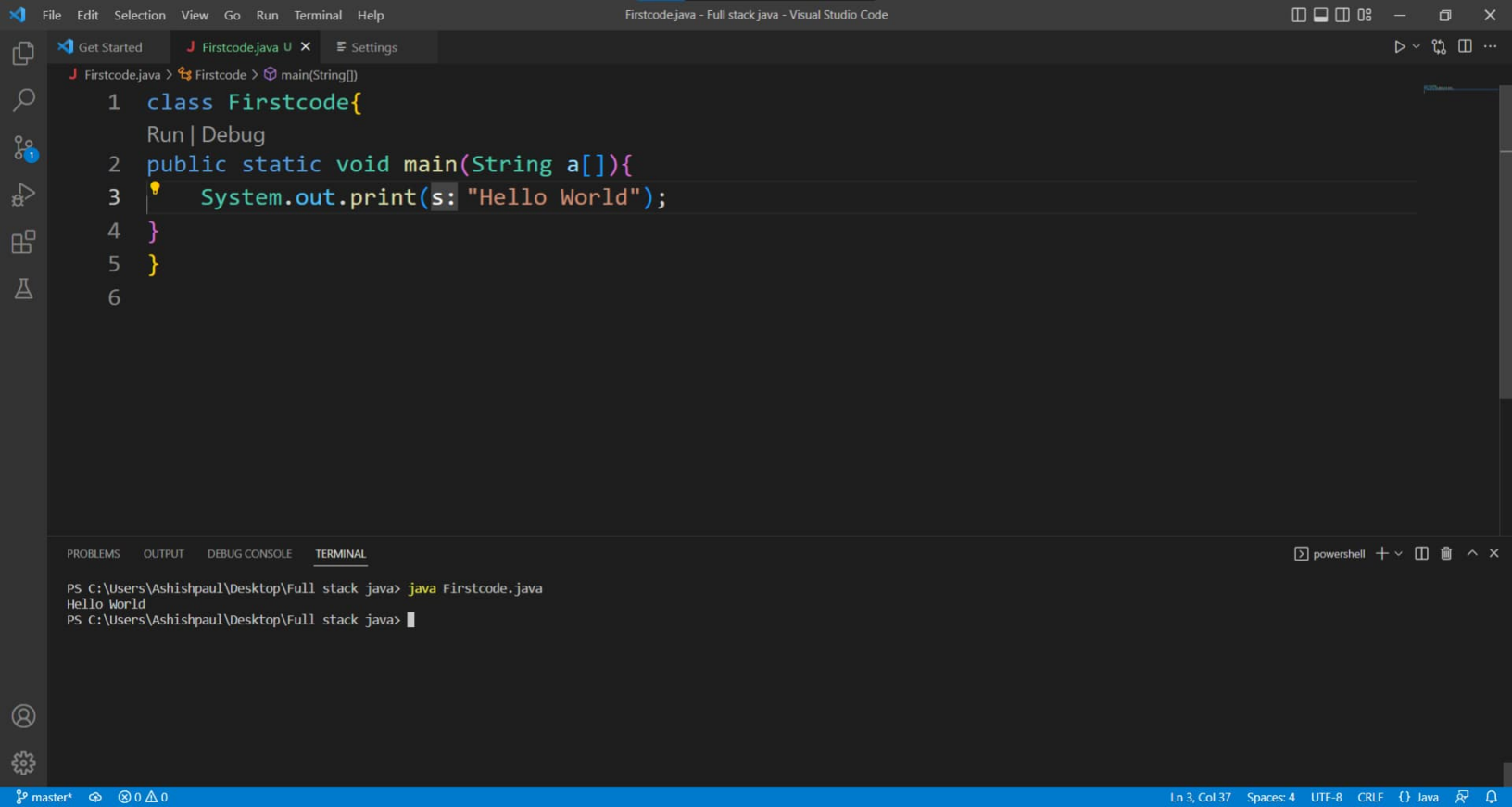
```
Class SecondCode
      ↳ Class name
{
    Public static Void main( ){
        System.out.Print("Hello");
    }
}
```

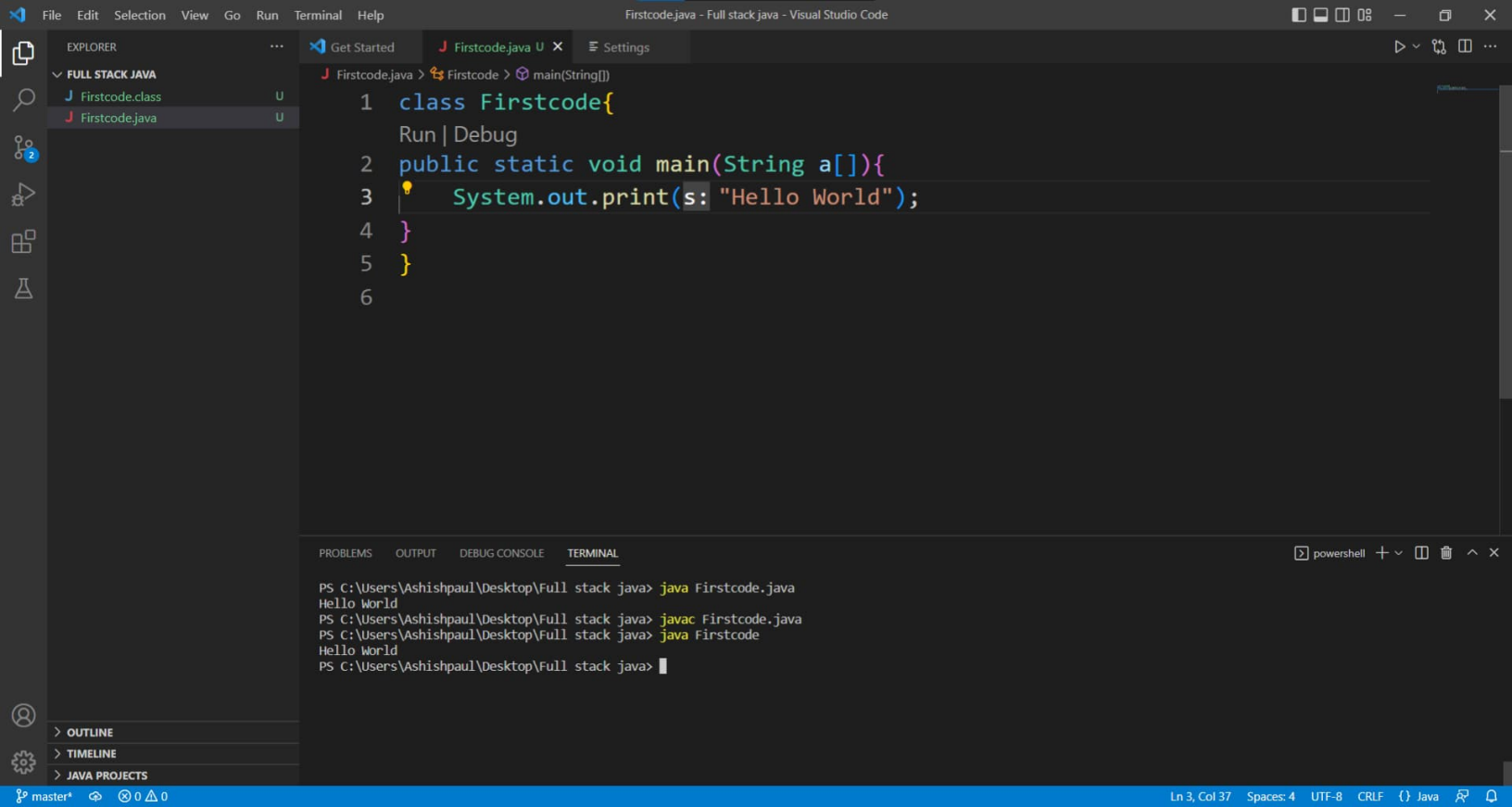
① Compile

Javac firstCode.Java

② Interpret

Java SecondCode.Ja


File Edit Selection View Go Run Terminal Help
Firstcode.java - Full stack java - Visual Studio Code
Get Started
Firstcode.java U
Settings
Firstcode.java > Firstcode > main(String[])
class Firstcode{
Run | Debug
public static void main(String a[]){
System.out.print(s: "Hello World");
}
}
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
powershell
PS C:\Users\Ashishpaul\Desktop\Full stack java> java Firstcode.java
Hello World
PS C:\Users\Ashishpaul\Desktop\Full stack java>
master*
Ln 3, Col 37 Spaces: 4 UTF-8 CRLF Java


File Edit Selection View Go Run Terminal Help
Firstcode.java - Full stack java - Visual Studio Code
EXPLORER
FULL STACK JAVA
Firstcode.class U
Firstcode.java U
Get Started
Firstcode.java U
Settings
Firstcode.java > Firstcode > main(String[])
1 class Firstcode{
Run | Debug
2 public static void main(String a[]){
3 System.out.print(s: "Hello World");
4 }
5 }
6
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
powershell
PS C:\Users\Ashishpaul\Desktop\Full stack java> java Firstcode.java
Hello World
PS C:\Users\Ashishpaul\Desktop\Full stack java> javac Firstcode.java
PS C:\Users\Ashishpaul\Desktop\Full stack java> java Firstcode
Hello World
PS C:\Users\Ashishpaul\Desktop\Full stack java>
OUTLINE
TIMELINE
JAVA PROJECTS
master*
0 0
Ln 3, Col 37 Spaces: 4 UTF-8 CRLF Java

(1) no Compulsion to have class name and file name same.

(2) if we make class public then filename and class name should be same.

→ FirstCode.Java – file name

→ Public class FirstCode {
  ↳ class name
  ↳ Class Made public

(3) In one file we can Create Multiple classes.

Eg: class SecondCode {

}

Class ThirdCode {

}.

→ Different packaging techniques

(1) .Exe : windows [all the file in folder and they Make it Executable.]

(2) .pkg : Mac

(3) .deb : linux

→ In Java we create .Jar files.

(4) .Jar : Java archive
  ↳ when open we get all class files.

Eg:- Package of class files.

→ Some Jar files are Executable.

100 - Java files
600 - classes
↳ Jar file