Language Basics

Sensitivity: Internal & Restricted

Objectives

In this module you will learn about :

- Path and Classpath
- Command line arguments
- Keywords
- Basic data types
- Types of Operators.

Language Basics

A Simple Java Program – The Set Up

- Before we learn about writing a simple java program, let us understand what preparations are needed for running a java program from console

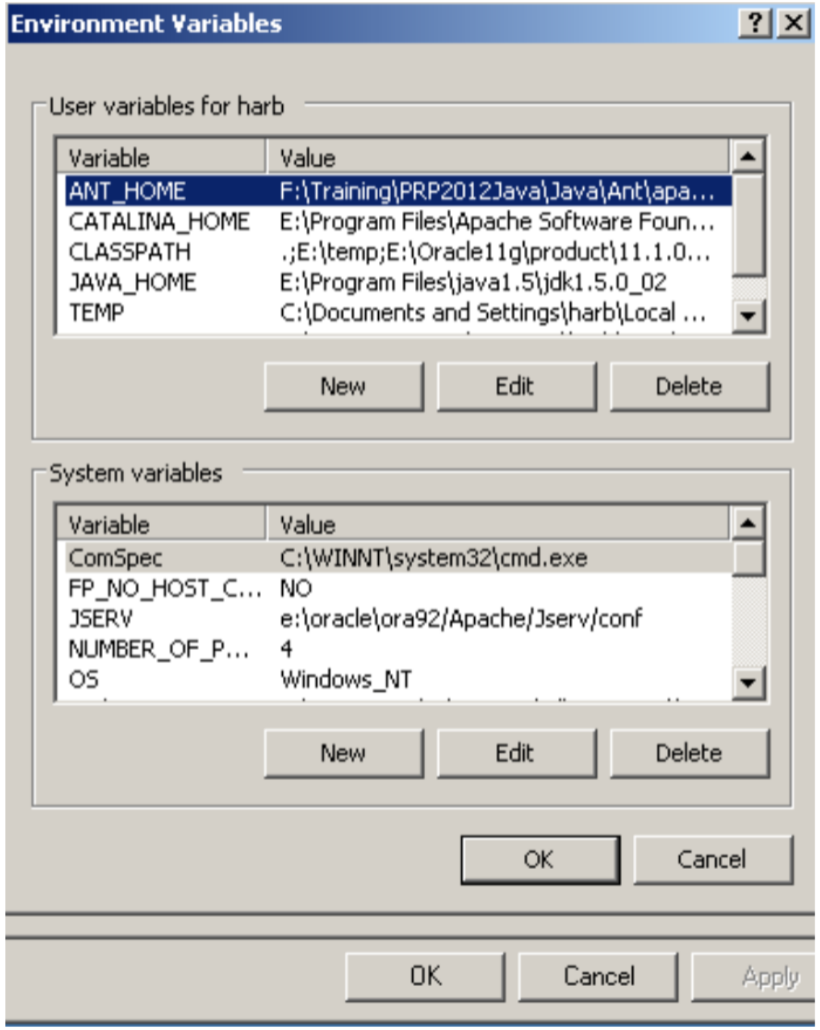
- We have to define PATH and CLASSPATH parameters and create necessary directories(folders) where we will be storing all our program files

PATH

- PATH is an *environmental variable* in DOS(Disk Operating System), Windows and other operating systems like Unix.
- PATH tells the operating system which directories(folders) to search for executable files, in response to commands issued by a user .
- It is a convenient way of executing files without bothering about providing the absolute path to the folder, where the file is located.

How to set PATH ?

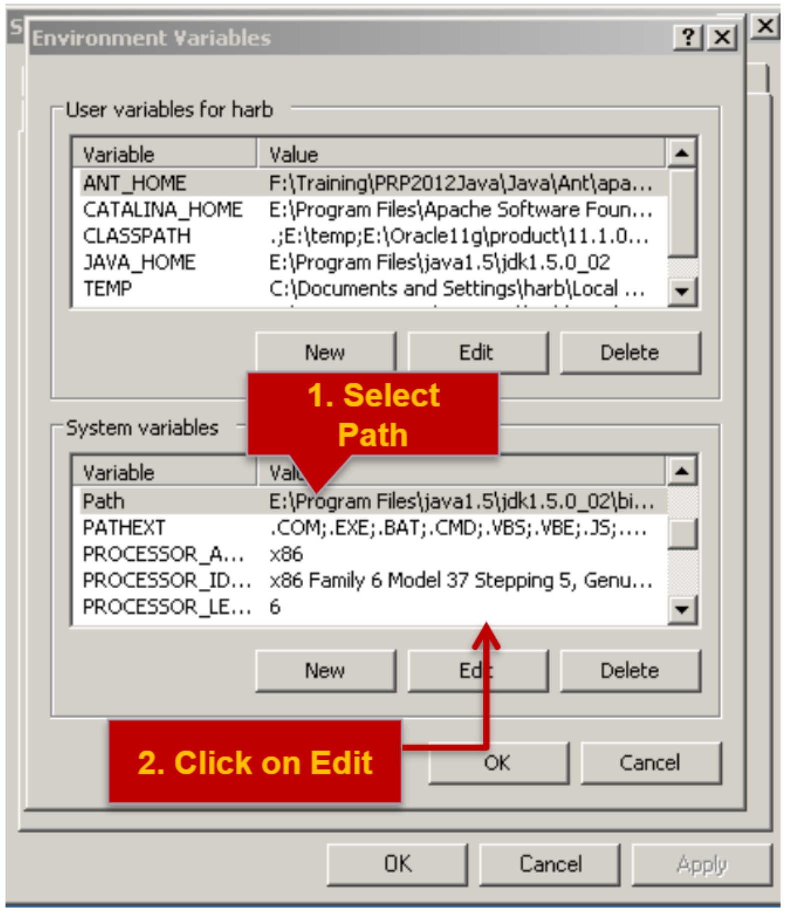
1. Right Click My Computer
2. Select Properties
3. You will get to see the Properties Page of My Computer
4. Select Advanced Tab
5. Select Environment Variables
6. You will see Environment Variables Page as displayed here



Variable	Value
ANT_HOME	F:\Training\PRP2012Java\Java\Ant\apa...
CATALINA_HOME	E:\Program Files\Apache Software Foun...
CLASSPATH	.;E:\temp;E:\Oracle11g\product\11.1.0...
JAVA_HOME	E:\Program Files\java1.5\jdk1.5.0_02
TEMP	C:\Documents and Settings\harb\Local ...

Variable	Value
ComSpec	C:\WINNT\system32\cmd.exe
FP_NO_HOST_C...	NO
JSERV	e:\oracle\ora92/Apache/Jserv/conf
NUMBER_OF_P...	4
OS	Windows_NT

How to set PATH ? (Contd.).



Environment Variables

User variables for harb

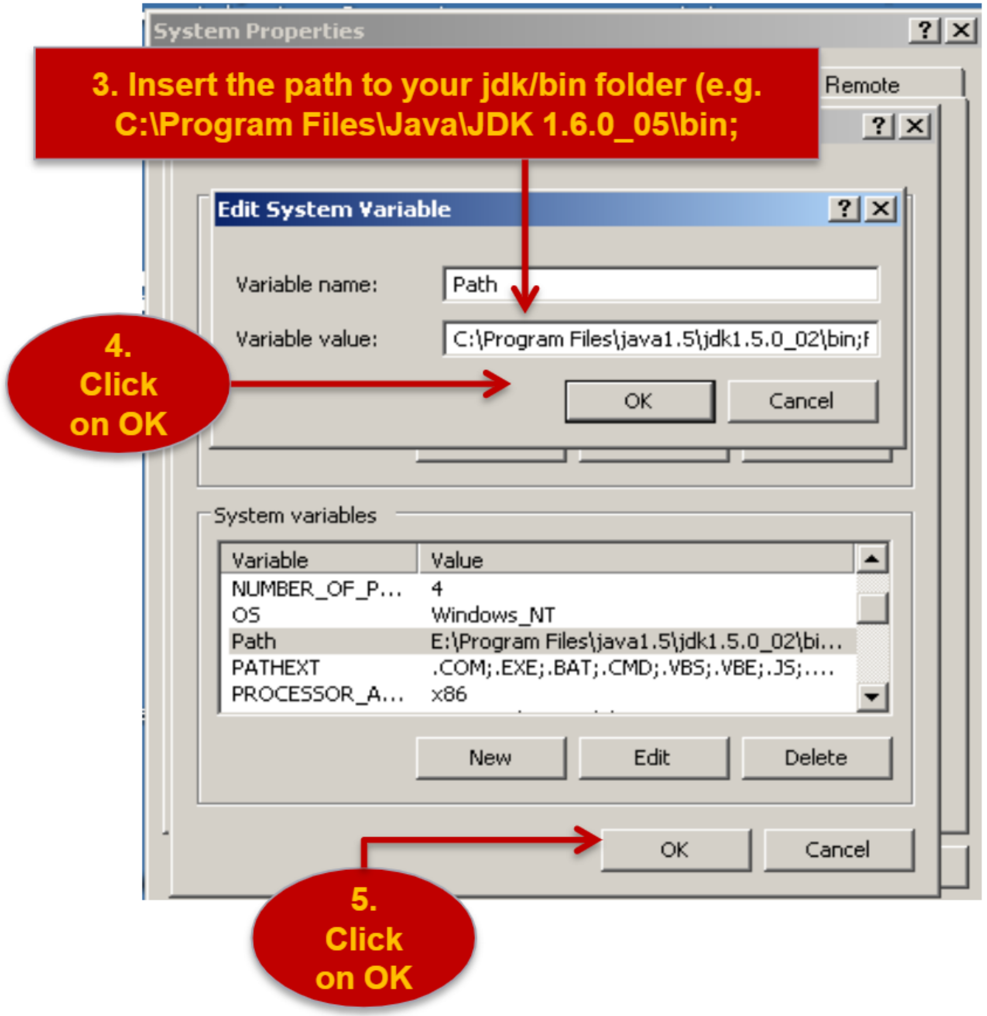
Variable	Value
ANT_HOME	F:\Training\PRP2012Java\Java\Ant\apa...
CATALINA_HOME	E:\Program Files\Apache Software Foun...
CLASSPATH	.;E:\temp;E:\Oracle11g\product\11.1.0...
JAVA_HOME	E:\Program Files\java1.5\jdk1.5.0_02
TEMP	C:\Documents and Settings\harb\Local ...

System variables

Variable	Value
Path	E:\Program Files\java1.5\jdk1.5.0_02\bi...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;....
PROCESSOR_A...	x86
PROCESSOR_ID...	x86 Family 6 Model 37 Stepping 5, Genu...
PROCESSOR_LE...	6

1. Select Path

2. Click on Edit



System Properties

Edit System Variable

Variable name: Path

Variable value: C:\Program Files\java1.5\jdk1.5.0_02\bin;f

System variables

Variable	Value
NUMBER_OF_P...	4
OS	Windows_NT
Path	E:\Program Files\java1.5\jdk1.5.0_02\bi...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;....
PROCESSOR_A...	x86

3. Insert the path to your jdk/bin folder (e.g. C:\Program Files\Java\JDK 1.6.0_05\bin;

4. Click on OK

5. Click on OK

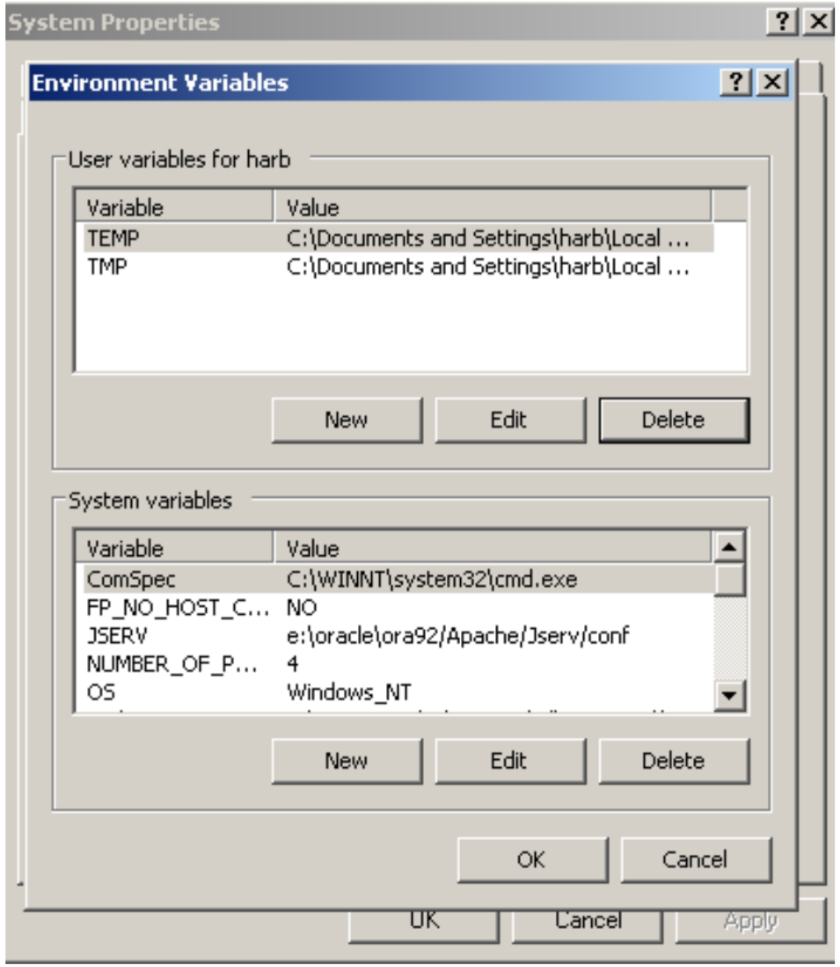
Sensitivity: Internal & Restricted

CLASSPATH

- CLASSPATH is a parameter which tells the JVM or the Compiler, where to locate classes that are not part of Java Development ToolKit(JDK).
- CLASSPATH is set either on command-line or through environment variable.
- CLASSPATH set on command-line is temporary in nature, while the environment variable is permanent.

How to set CLASSPATH ?

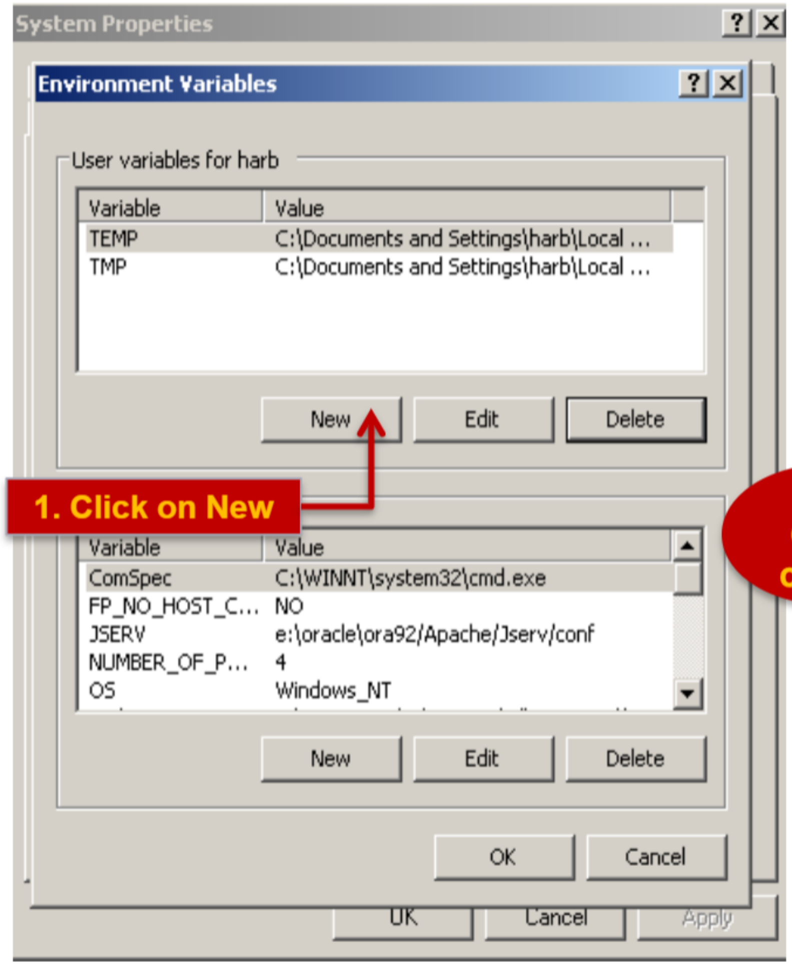
1. Right Click My Computer
2. Select Properties
3. You will get to see the Properties Page of My Computer
4. Select Advanced Tab
5. Select Environment Variables
6. You will see Environment Variables Page as displayed here



Variable	Value
TEMP	C:\Documents and Settings\harb\Local ...
TMP	C:\Documents and Settings\harb\Local ...

Variable	Value
ComSpec	C:\WINNT\system32\cmd.exe
FP_NO_HOST_C...	NO
JSERV	e:\oracle\ora92/Apache/Jserv/conf
NUMBER_OF_P...	4
OS	Windows_NT

How to set CLASSPATH ? (Contd.).

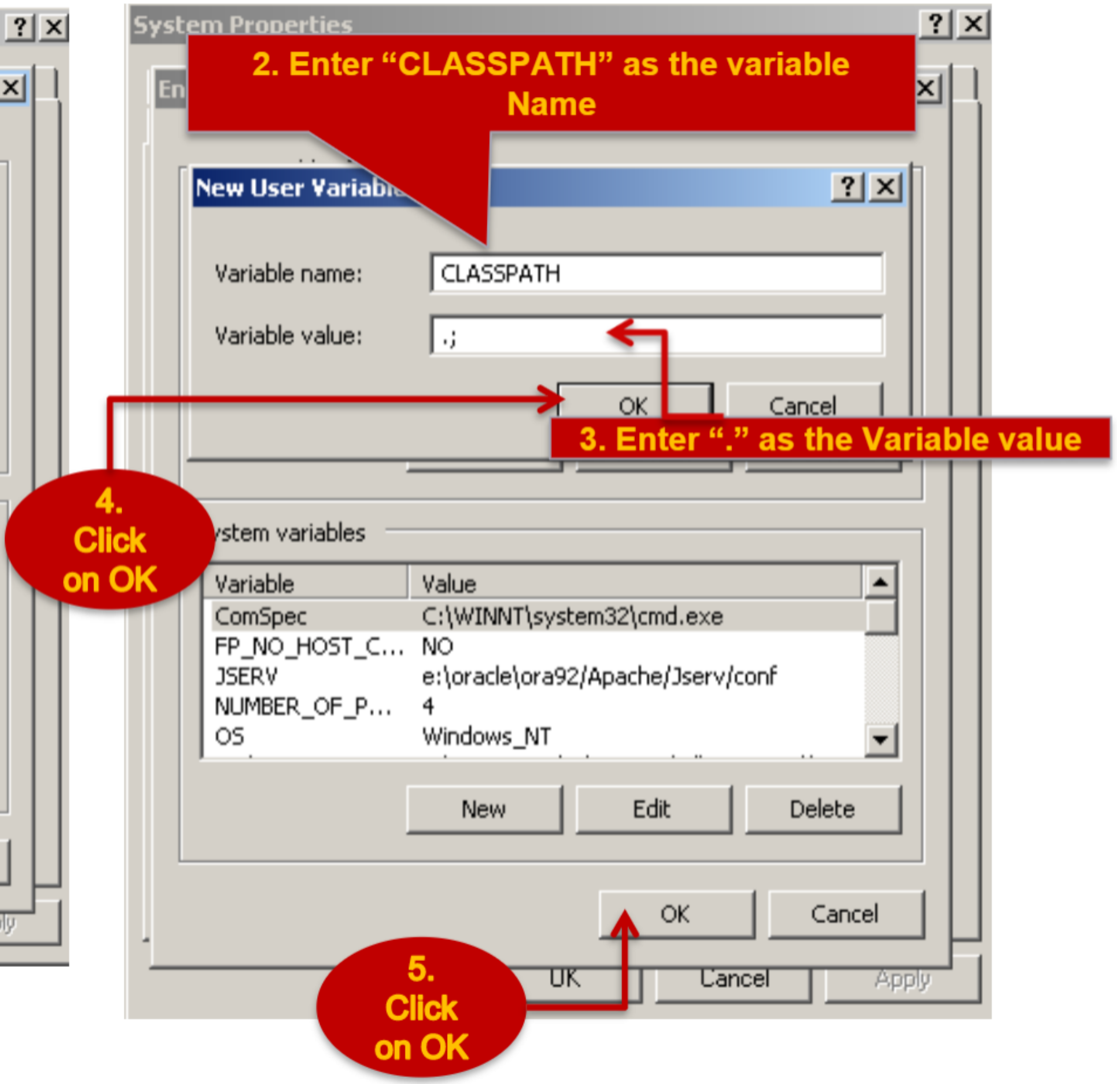


User variables for harb

Variable	Value
TEMP	C:\Documents and Settings\harb\Local ...
TMP	C:\Documents and Settings\harb\Local ...

Variable	Value
ComSpec	C:\WINNT\system32\cmd.exe
FP_NO_HOST_C...	NO
JSERV	e:\oracle\ora92/Apache/Jserv/conf
NUMBER_OF_P...	4
OS	Windows_NT

1. Click on New



Variable name: CLASSPATH

Variable value: .;

System variables

Variable	Value
ComSpec	C:\WINNT\system32\cmd.exe
FP_NO_HOST_C...	NO
JSERV	e:\oracle\ora92/Apache/Jserv/conf
NUMBER_OF_P...	4
OS	Windows_NT

2. Enter “CLASSPATH” as the variable Name

3. Enter “.” as the Variable value

4. Click on OK

5. Click on OK

A Simple Java Program

Our first Java Program:

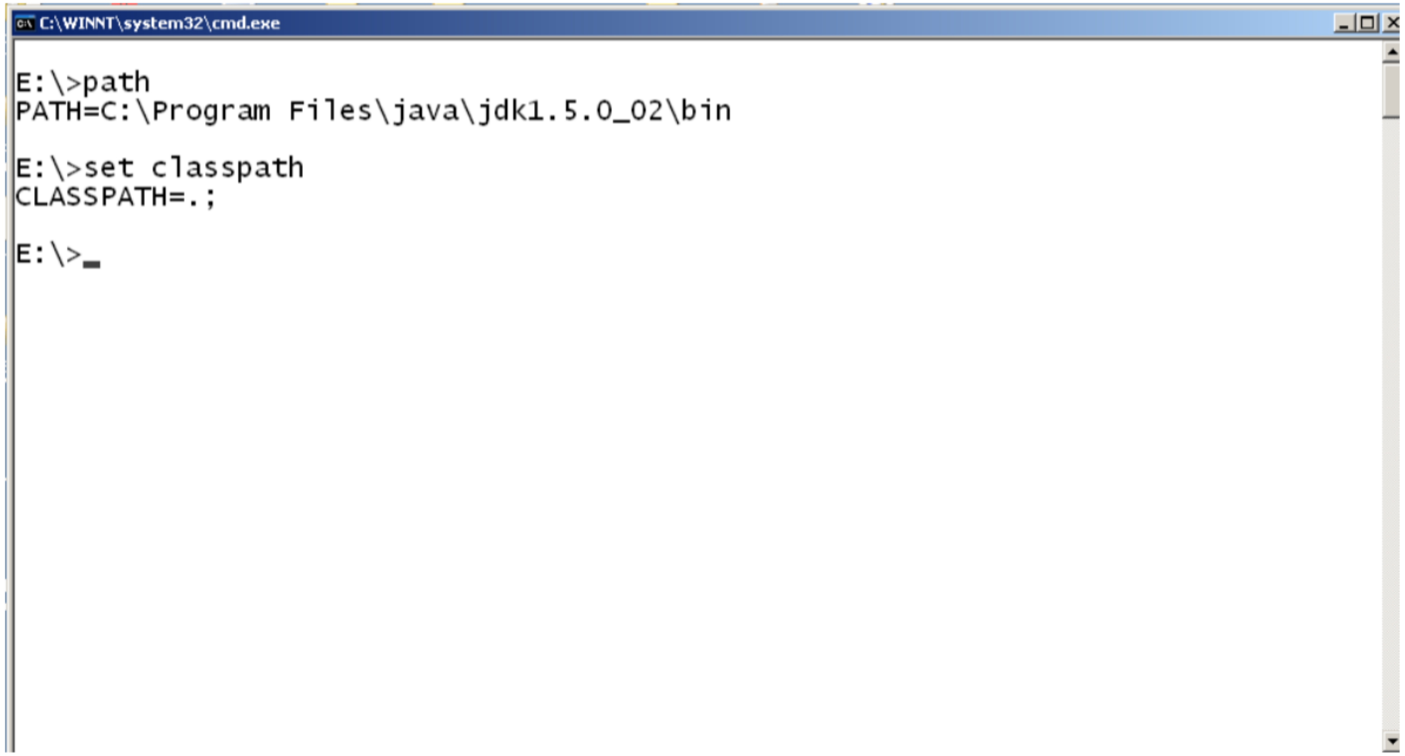
```
public class Welcome {
     public static void main(String args[]) {
          System.out.println("Welcome..!");
    }
}
```

This program displays the output "Welcome..!" on the console

Create source file	:	Welcome.java
Compile	:	javac Welcome.java
Execute	:	java Welcome

Executing your first Java Program

- Before executing the program, just check whether the PATH and the CLASSPATH parameters are properly set, by typing in the commands as shown in the screen below:

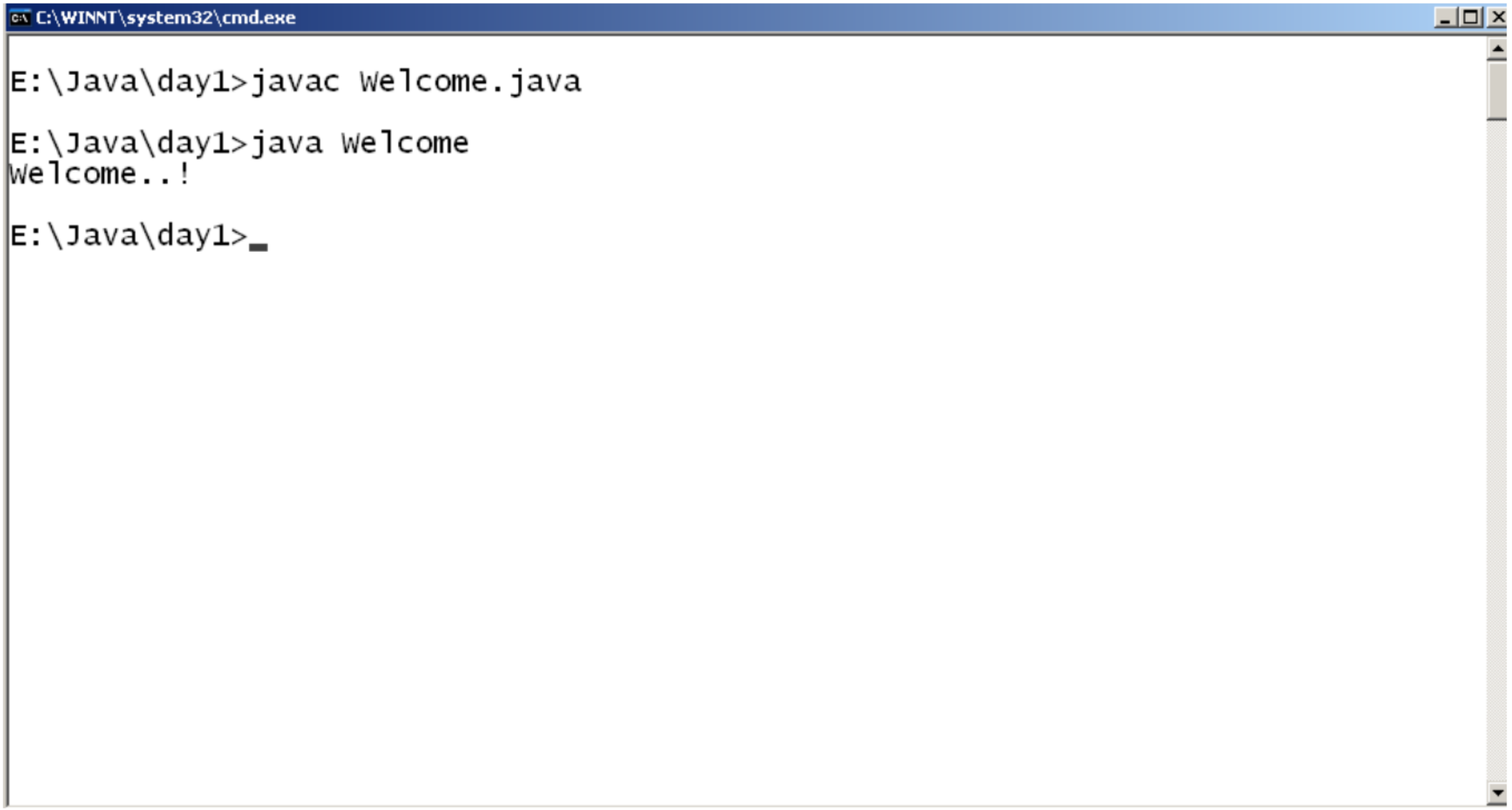


```
E:\>path
PATH=C:\Program Files\java\jdk1.5.0_02\bin

E:\>set classpath
CLASSPATH=.;

E:\>_
```

Sensitivity: Internal & Restricted

Executing your first Java Program (Contd.).

- Now compile and execute your program as given below :



```
E:\Java\day1>javac Welcome.java

E:\Java\day1>java Welcome
Welcome..!

E:\Java\day1>_
```

Quiz

Sample.java file contains class A, B and C. How many .class files will be created after compiling Sample.java ? What is your observation ?

Sample.java

```
class A {
    void m1() { }
}
class B {
    void m2() { }
}
class C {
    void m3() { }
}
```

Quiz(Contd..)

What will be the result if you try to compile and execute the following program ?

Reason out :

Sample.java

```
class Sample {
   public static void main() {
    System.out.println("Welcome");
   }
}
```

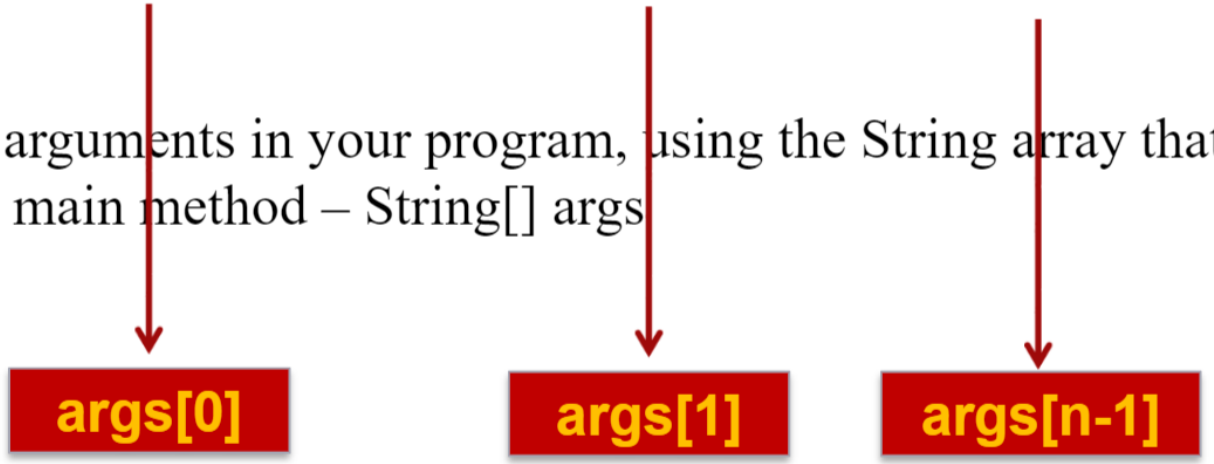
a. Compilation Error
b. Runtime Error
c. The program compiles and executes successfully but prints nothing.
d. It will print "Welcome"

Accessing command line arguments

When you execute a java program, you can pass command line arguments in the following way :

```
java Simple <argument1> <argument2>….<argument-n>
```

You can access these arguments in your program, using the String array that you have passed as an argument to the main method – String[] args



Passing command line arguments

```
class Simple {
  static public void main(String[] args) {
    System.out.println(args[0]);
  }
}
```

When we compile the above code successfully and execute it as Java Simple Wipro, the output will be :

Wipro

Accessing numeric command line arguments

```
class Simple {
  static public void main(String[] args) {
    int i1 = Integer.parseInt(args[0]);
    int i2 = Integer.parseInt(args[1]);
    System.out.println(i1+i2);
  }
}
```

When we compile the above code successfully and execute it as Java Simple 10 20, the output will be :

30

Finding length of an Array

- How to find the number of command line arguments that a user may pass while executing a java program?
- The answer to the above question lies in **args.length**, where args is the String array that we pass to the main method and length is the property of the Array Object

Example on Finding length of an Array

```
class FindNumberOfArguments {
    public static void main(String[ ] args) {
        int len = args.length;
        System.out.println(len);
    }
}
```

If you compile the above code successfully and execute it as java FindLength A B C D E F, the result will be

6

And if you execute it as java FindLength Tom John Lee, the result will be

3