Implementing A JAVA Program

Implementation of a Java application program involves a following step. They include:

1. Creating the program
2. Compiling the program
3. Running the program

Remember that, before we begin creating the program, the Java Development Kit (JDK) must be properly installed on our system and also path will be set.

- **Creating Program**

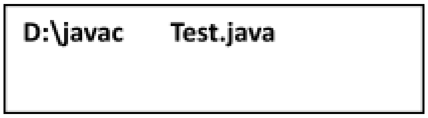
  We can create a program using Text Editor (Notepad) or IDE (NetBeans)

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

  File Save : d:\Test.java

- Compiling the program

  To compile the program, we must run the Java compiler (javac), with the name of the source file on “**command prompt**” like as follows

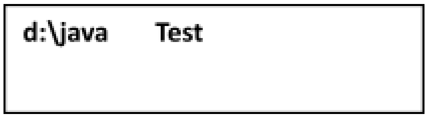


```
D:\javac     Test.java
```

  If everything is OK, the “**javac”** compiler creates a file called “**Test.class”** containing byte code of the program.

- Running the program

  We need to use the Java Interpreter to run a program.



```
d:\java     Test
```

Now the interpreter looks for the main method in the program and begins execution from there. After execution program successfully the output will be as follows,

Output:

```
My First Java Program.
```

Lets understand what above program : ---

class : class keyword is used to declare classes in Java

public : It is an access specifier. **Public** means this function is visible to all.

static : static is again a keyword used to make a function static. To execute a static function you do not have to create an Object of the class. The **main()** method here is called by JVM, without creating any object for class.

void : It is the return type, meaning this function will not return anything.

main : main() method is the most important method in a Java program. This is the method which is executed, hence all the logic must be inside the main() method. If a java class is not having a main() method, it causes compilation error.

String[] args : This is used to signify that the user may opt to enter parameters to the Java Program at command line. We can use both String[] args or String args[]. Java compiler would accept both forms.

System.out.println : This is used to print anything on the console like “**printf**” in C language.

CLASSPATH

1. CLASSPATH describes a location where all required files are available which is used in our application.
2. Java compiler and JVM will use CLASSPATH to locate required files.
3. If we do not set CLASSPATH then Java compiler will not able to find required files hence you will get error.

PATH

1. PATH variable is set to provide path for all java tools like javac, java, appletviewer.
2. PATH describes a location where binary executables are available.
3. If we do not set PATH then our system will not be able to find where javac is; hence it will not work. It is mandatory to set path.

How to set path in Java

If you are saving the java source file inside the jdk/bin directory, path is not required to be set because all the tools will be available in the current directory.

But If you are having your java file outside the jdk/bin folder, it is necessary to set path of JDK.

There are 2 ways to set java path:

1. Temporary
2. Permanent

1) How to set Temporary Path of JDK in Windows

To set the temporary path of JDK, you need to follow following steps:

- Open command prompt
- copy the path of jdk/bin directory
- write in command prompt: set path=copied_path

For Example:

set path=C:\Program Files\Java\jdk1.7.0_15\bin

2) How to set Permanent Path of JDK in Windows 7

For setting the permanent path of JDK, you need to follow these steps:

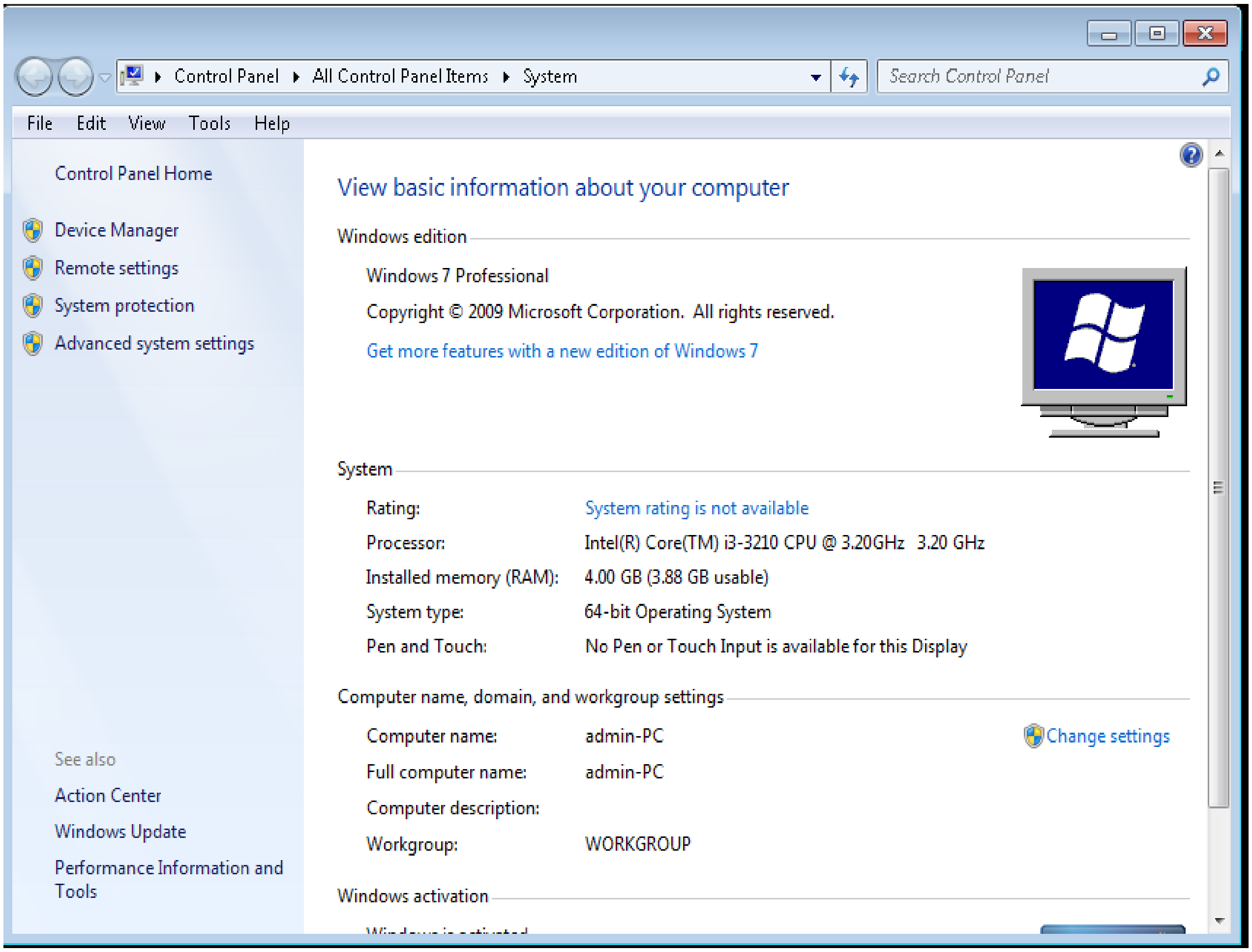
Go to MyComputer properties ->Click on Advanced system settings -> select Advance Tab ->click on environment variables -> click on new button of user variable

write

Variable name: path

Variable Value: path of bin folder (eg: c:\Program Files\Java\jdk1.8_51\bin;.;)

-> write path of bin folder in variable value -> ok -> ok -> ok



View basic information about your computer

Windows edition

Windows 7 Professional

Copyright © 2009 Microsoft Corporation. All rights reserved.

Get more features with a new edition of Windows 7

System

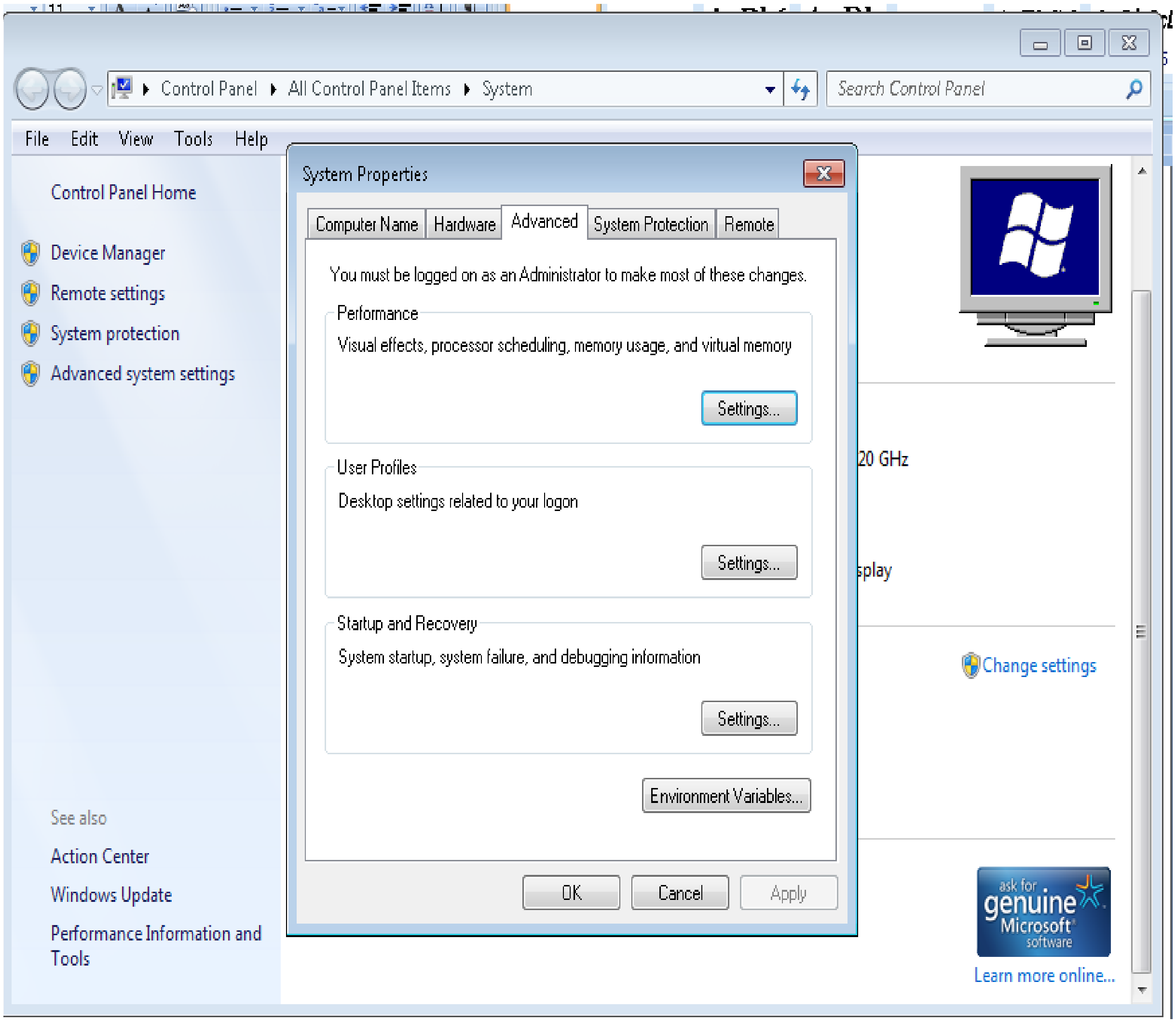
Rating:	System rating is not available
Processor:	Intel(R) Core(TM) i3-3210 CPU @ 3.20GHz 3.20 GHz
Installed memory (RAM):	4.00 GB (3.88 GB usable)
System type:	64-bit Operating System
Pen and Touch:	No Pen or Touch Input is available for this Display

Computer name, domain, and workgroup settings

Computer name:	admin-PC
Full computer name:	admin-PC
Computer description:	
Workgroup:	WORKGROUP

Windows activation

Windows is activated

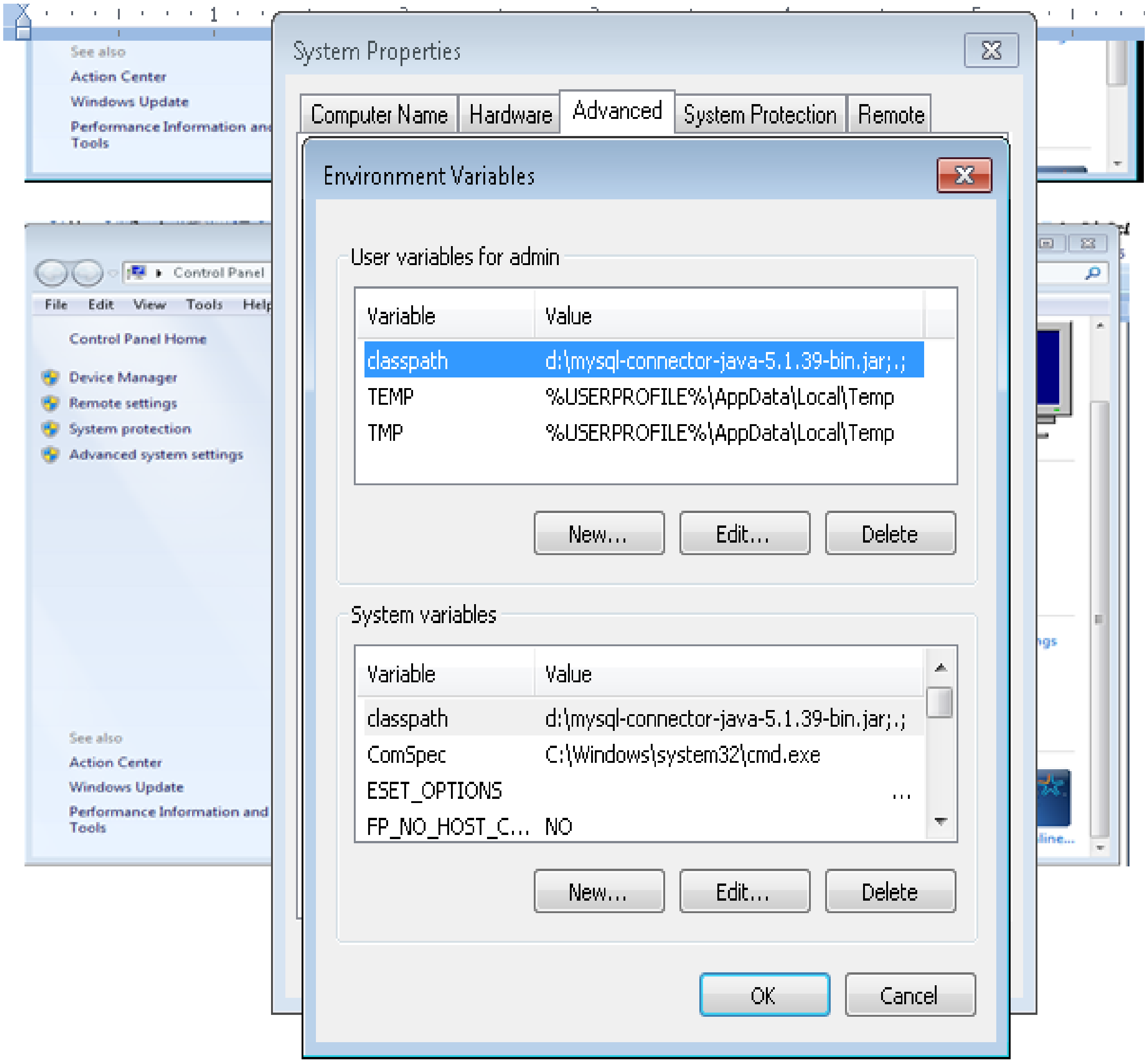


You must be logged on as an Administrator to make most of these changes.

Visual effects, processor scheduling, memory usage, and virtual memory

Desktop settings related to your logon

System startup, system failure, and debugging information

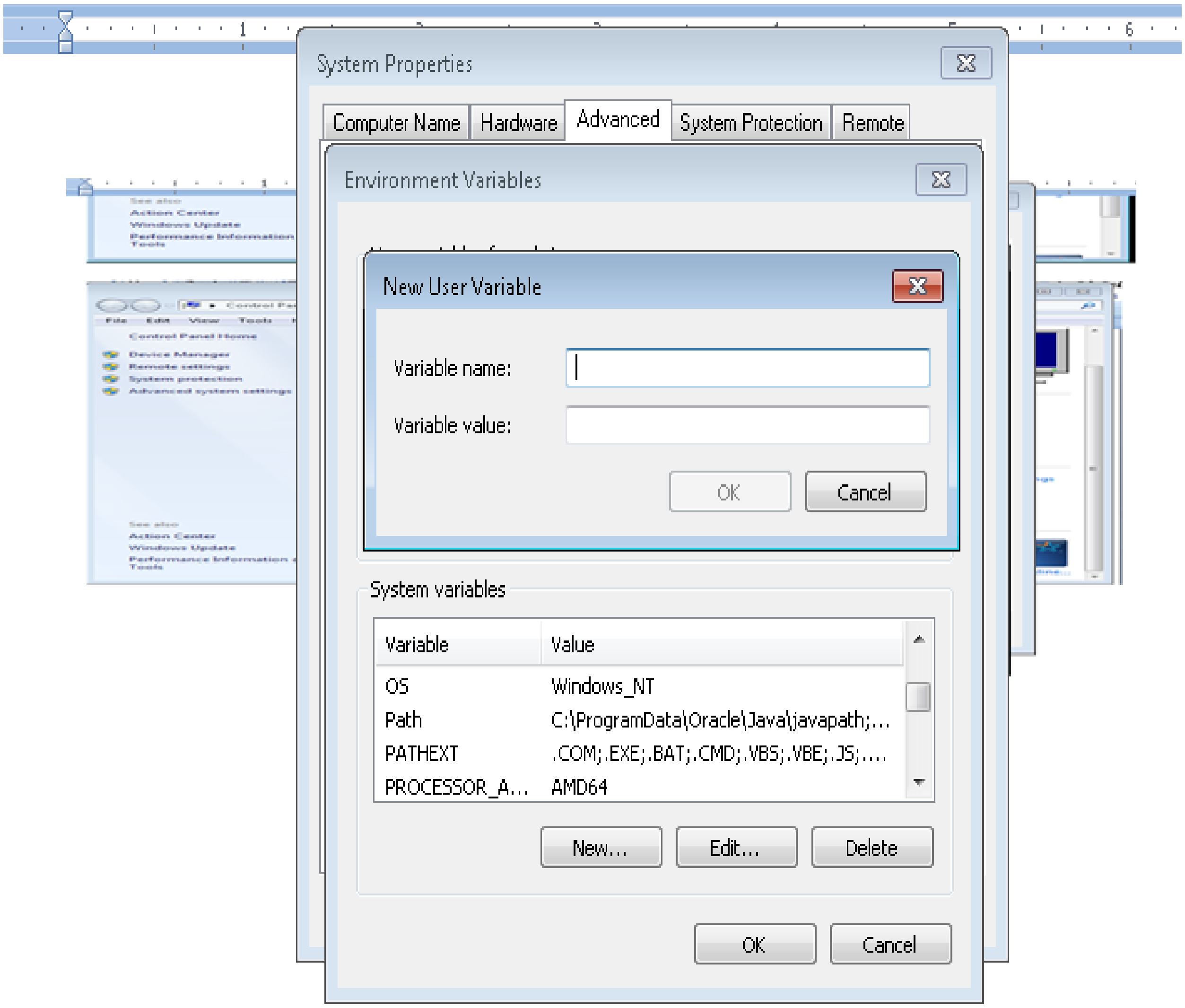


User variables for admin

Variable	Value
classpath	d:\mysql-connector-java-5.1.39-bin.jar;.;
TEMP	%USERPROFILE%\AppData\Local\Temp
TMP	%USERPROFILE%\AppData\Local\Temp

System variables

Variable	Value
classpath	d:\mysql-connector-java-5.1.39-bin.jar;.;
ComSpec	C:\Windows\system32\cmd.exe
ESET_OPTIONS	...
FP_NO_HOST_C...	NO



Variable	Value
OS	Windows_NT
Path	C:\ProgramData\Oracle\Java\javapath;...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;....
PROCESSOR_A...	AMD64

Variable Name: path

Under Variable value: c:\Program Files\Java\jdk1.8_51\bin; . ;

Q: Write a Java program to calculate the sum and product of two given number.

```
class Sample {
   public static void main(String args[])
   {
      int a,b,sum=0,product=0;
      a=10;
      b=20;
      sum=a+b;
      product=a*b;
      System.out.println("Sum="+sum);
      System.out.println("Product="+product);
   }
};
```

Q: Write a Java program to accept two numbers from the user and calculate sum and product.

```
import java.util.*;
  class Sample {
   public static void main(String args[])
   {
      int a,b,sum=0,product=0;
      Scanner scan=new Scanner(System.in);
      System.out.println("Enter the First No:");
      a=scan.nextInt();
      System.out.println("Enter the Second No:");
      b=scan.nextInt();
      sum=a+b;
      product=a*b;
```

```
        System.out.println("Sum="+sum);
        System.out.println("Product="+product);
    }
}
```

Q: Write a Java program to accept two number from the user and calculate the average.

```
import java.util.*;
public class Sample {
    public static void main(String args[])
    {
       double a,b,sum=0,avg=0;
        Scanner scan=new Scanner(System.in);
        System.out.println("Enter the First No:");
        a=scan.nextDouble();
        System.out.println("Enter the Second No:")
        b=scan.nextDouble();
        sum=a+b;
        avg=sum/2;
        System.out.println("Average="+avg);
    }
}
```

Q: Write a java program to convert fahrenheit to celsius and celsius to Fahrenheit using formula c = (5.0/9.0)*(f-32).

```
import java.util.Scanner;
public class Fahrenheit_Celsius
{
    public static void main(String[] args)
    {
        double c, f;
        Scanner s = new Scanner(System.in);
        System.out.print("Enter temperature in Fahrenheit:");
        f = s.nextDouble();
        c = (5.0/9.0)*(f-32);
        System.out.println("Temperature in Celsius:"+c);
    }
}
```

Output:

```
Enter temperature in Fahrenheit:15
Temperature in Celsius:-9.444444444444445
```

Q: Write a Java program to calculate the area of triangle using three sides.

```
    import java.lang.*;
import java.util.Scanner;
public class Area
{
  public static void main(String[] args)
  {

    int a,b,c;
    double s,area;
    Scanner scan = new Scanner(System.in);
    System.out.print("Enter the three sides:");
    a=scan.nextInt();
    b=scan.nextInt();
    c=scan.nextInt();
    s=(a+b+c)/2;
    area=Math.sqrt(s*(s-a)*(s-b)*(s-c));
    System.err.println("Area of Triangle:"+area);
 }
}
```

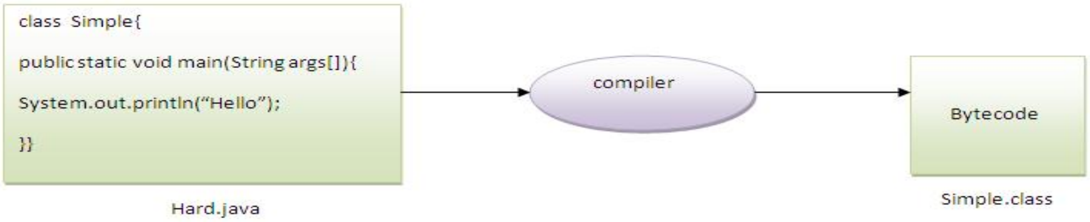
Q: Write a Java program to calculate the given mathematical equation.

- **b^2 _ 4ac**

```java
// Restrict the number of digits after decimal point in Java.
// formation the decimal values using the DecimalFormat class.



import java.util.Scanner;
import java.text.DecimalFormat;
public class Sample
{
  public static void main(String[] args)
  {
    int a,b,c;
    double s,area;
    Scanner scan = new Scanner(System.in);

    System.out.print("Enter the three sides:");
    a=scan.nextInt();
    b=scan.nextInt();
    c=scan.nextInt();
    s=(a+b+c)/2;
    area=Math.sqrt(s*(s-a)*(s-b)*(s-c));

    DecimalFormat fmt=new DecimalFormat("0.00");
    System.err.println("Area of Triangle:"+fmt.format(area));
  }
}
```

Can you save a java source file by other name than the class name?

Yes, if the class is not public



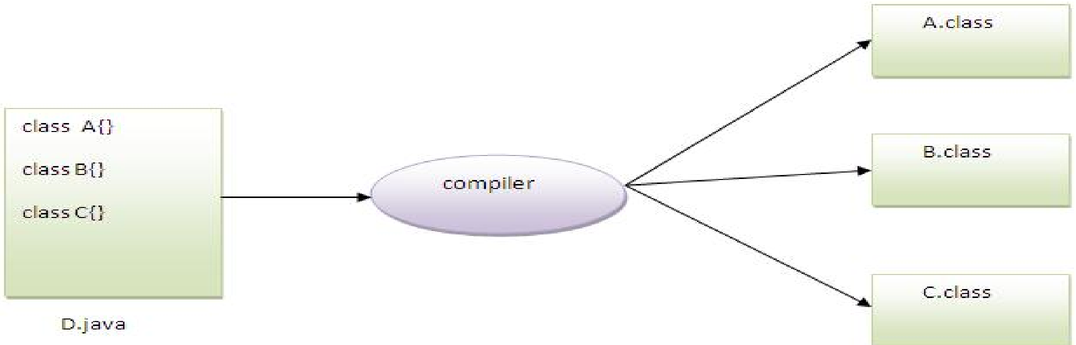
```
class  Simple{
public static void main(String args[]){
System.out.println(“Hello”);
}}
```

To compile: javac Hard.java

To execute: java Simple

Can you have multiple classes in a java source file?

Yes.



```
class  A{}
class B{}
class C{}
```