JAVA PROGRAMMING LAB
(MCA 4241)

Rules & Regulations

- Seating Arrangements
- Login Book Entry
- Mobiles in Class & Lab – Switch off mode.
- Lab Attendance- 85%
  - Totally 12 Labs including Final Lab exam.

Rules & Regulations

- Managing missing Lab.
  - Work extra time to complete the exercises and submit Lab Records.
  - Inform faculty well before –Personally/ email regarding Absence.

Lab Evaluation

- **Total Internal Evaluation – 60 Marks**
  - 2 Evaluations of 20 marks
    - Record : 6 marks
    - Exe-check: 7 marks
    - Quiz: 7 marks
  - 1 Mid-Term evaluation of 20 marks (approx. 6th lab)
    Write-up + execution

Lab End Semester Exam – 40 Marks

References

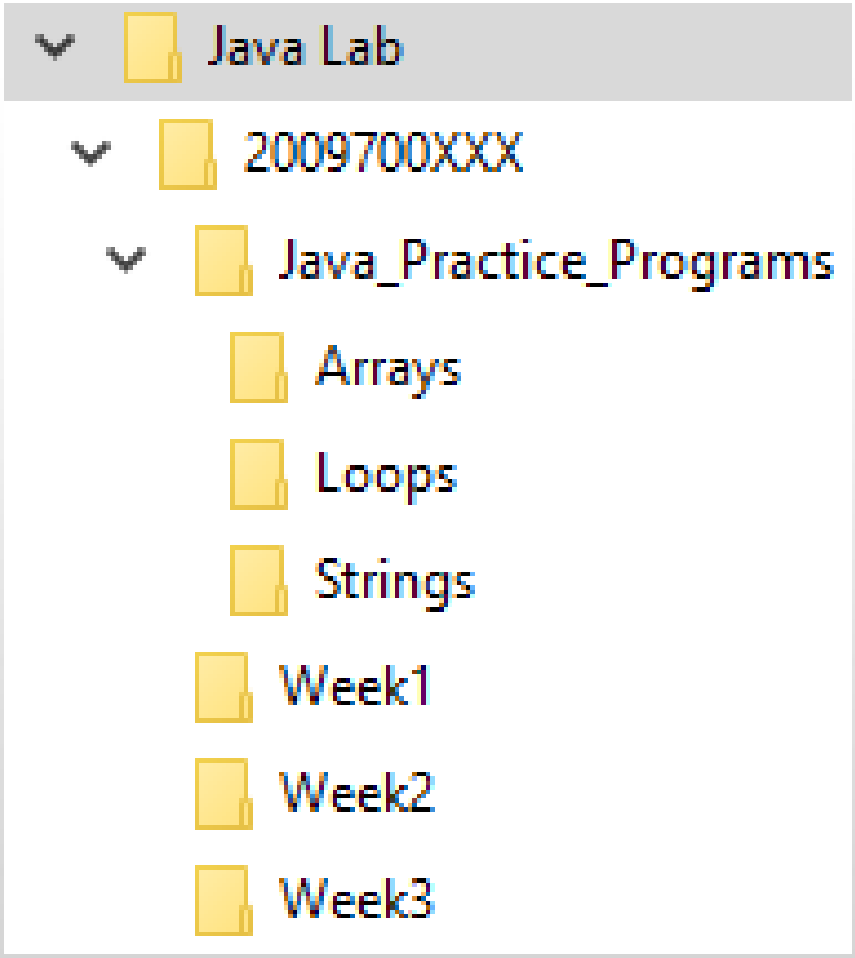
1. Herbert Schildt, *Java The Complete Reference*, 11th Edition, McGraw Hill, 2019.
2. Cay S. Horstmann. *Core Java: Volume I - Fundamentals*. 11th Edition, Pearson Education, 2018.
3. Cay S.Horstmann, *Core Java: Volume II – Advanced Features*, 11th Edition, Pearson Education, 2019.
4. Herbert Schildt and Dale Skrien, *Java Fundamentals*, Tata McGraw-Hill Education, 2015.

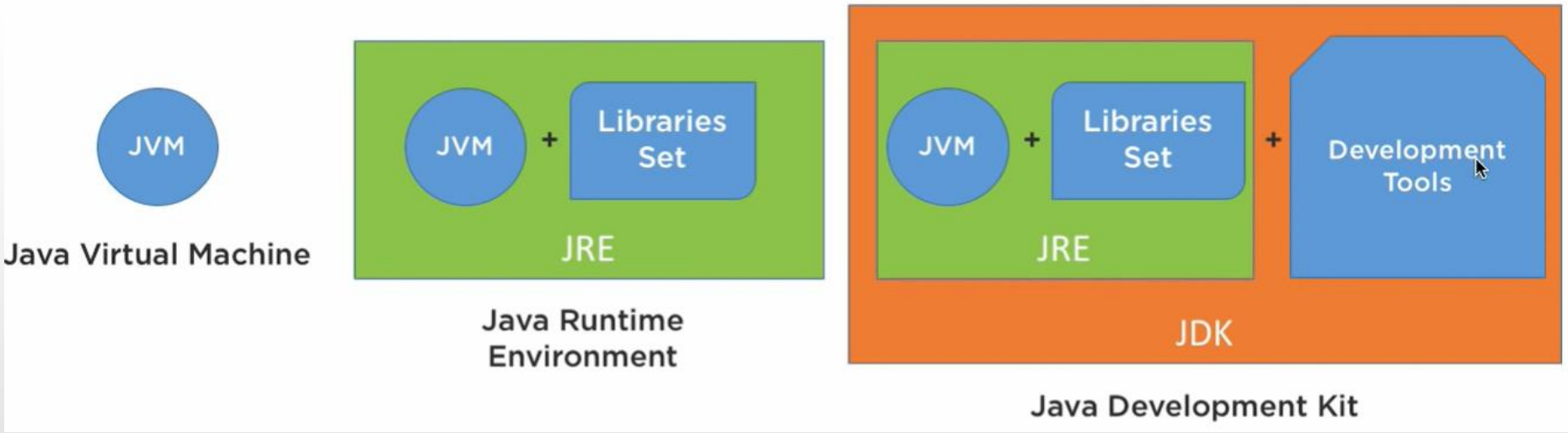
Course Outcomes

1. Developing Java Applications based on basic programming concepts.
2. Understand the object oriented concepts of java.
3. Implement programs with other object oriented concepts such as – Inheritance, multithreading.
4. Read/Write using Java streams and error handling
5. Design GUI components with the Java Swings

Saving Your Programs

Create folder with your Registration Number in D:\





PresentationPoint

Java Development Kit (JDK)

Java Runtime Environment (JRE)

Java Virtual Machine (JVM)

Jre = jvm + library classes

Jdk = jre + development tools

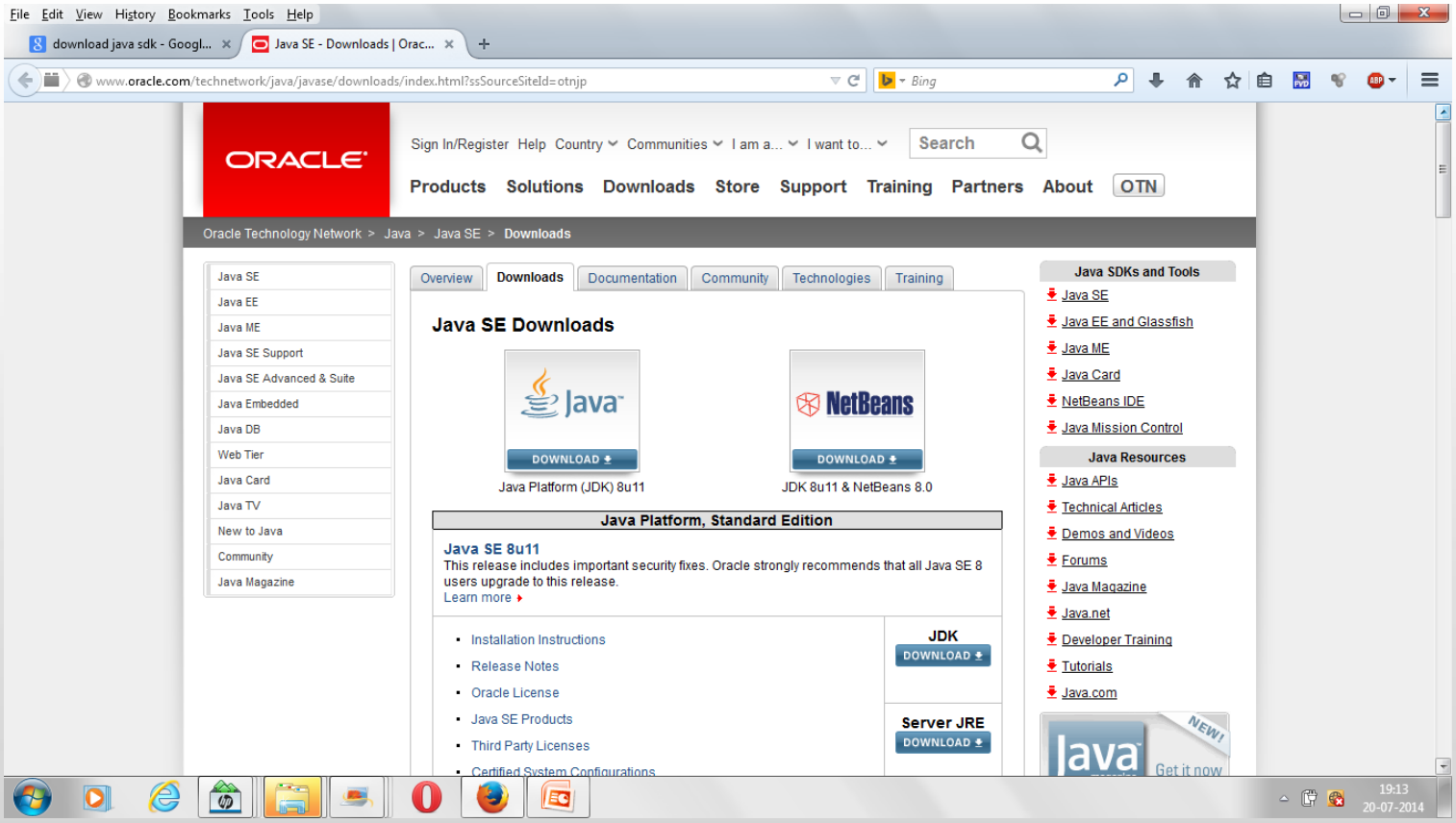
Example 1

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

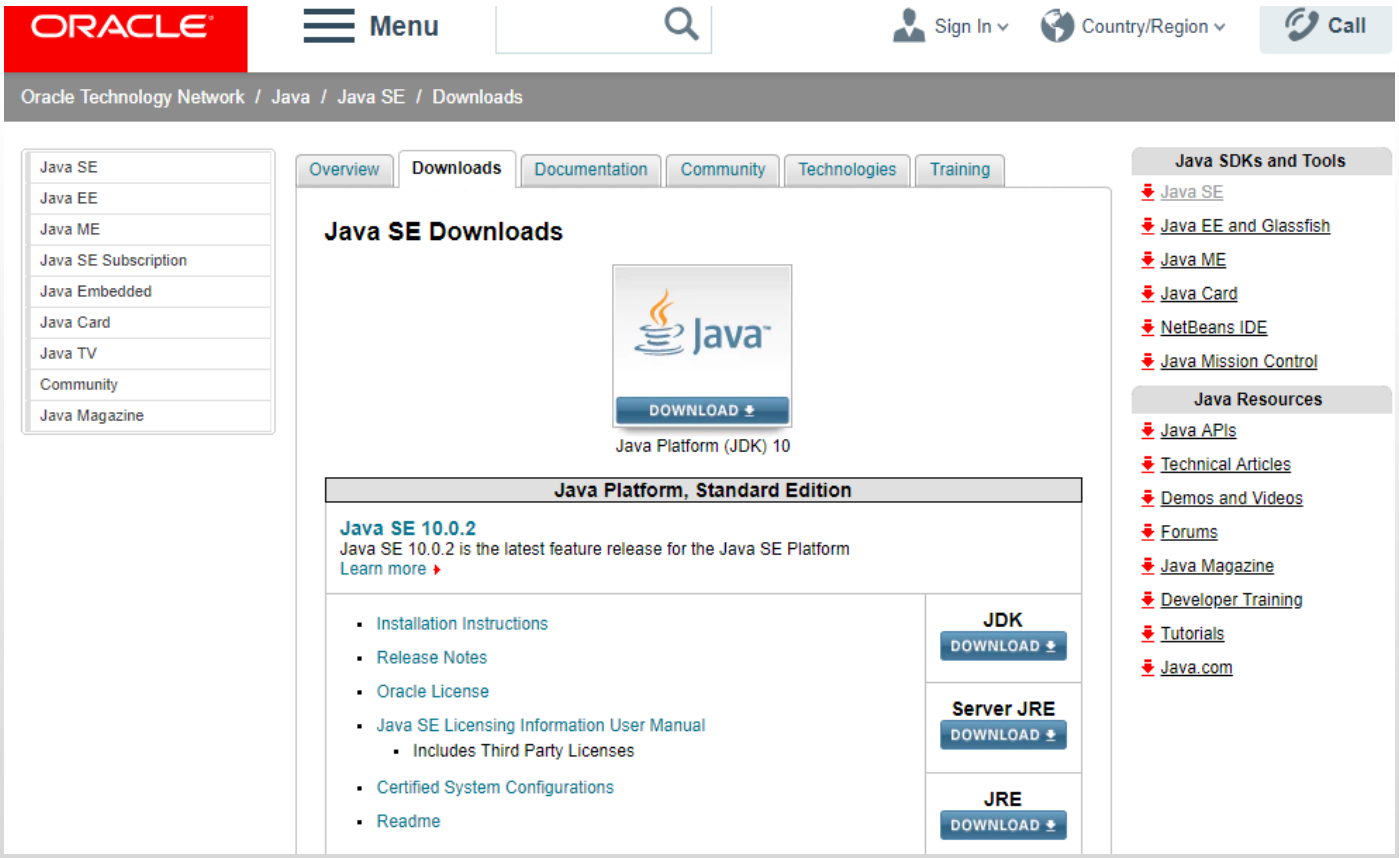
Note : String, System

class name is same as file name with the extension .java

Downloading and installing JDK



Downloading and installing JDK



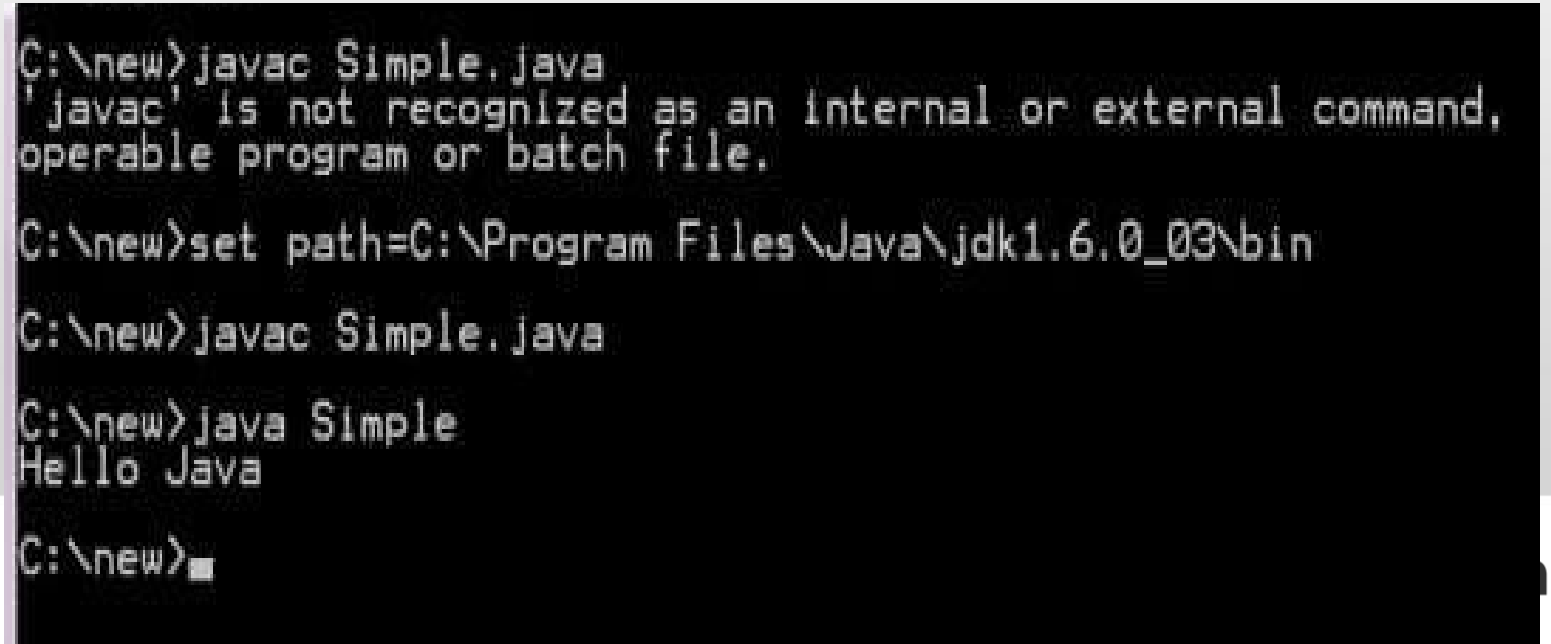
How to set path in Java

The path is required to be set for using tools such as javac, java, etc.

How to set the Temporary Path of JDK in Windows

- Open the command prompt
- Copy the path of the JDK/bin directory
- Write in command prompt: set path=copied_path

- For Example:
  - **set path=C:\Program Files\Java\jdk1.6.0_23\bin**



2) How to set Permanent Path of JDK in Windows

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

Go to My Computer properties →advanced tab
→ environment variables
→ new tab of user variable
→ Edit the path variable
→ write path of bin folder in variable value
→ ok → ok → ok

Now your permanent path is set.

You can now **execute any program of java from any drive**.

Compiling & Executing Java Program

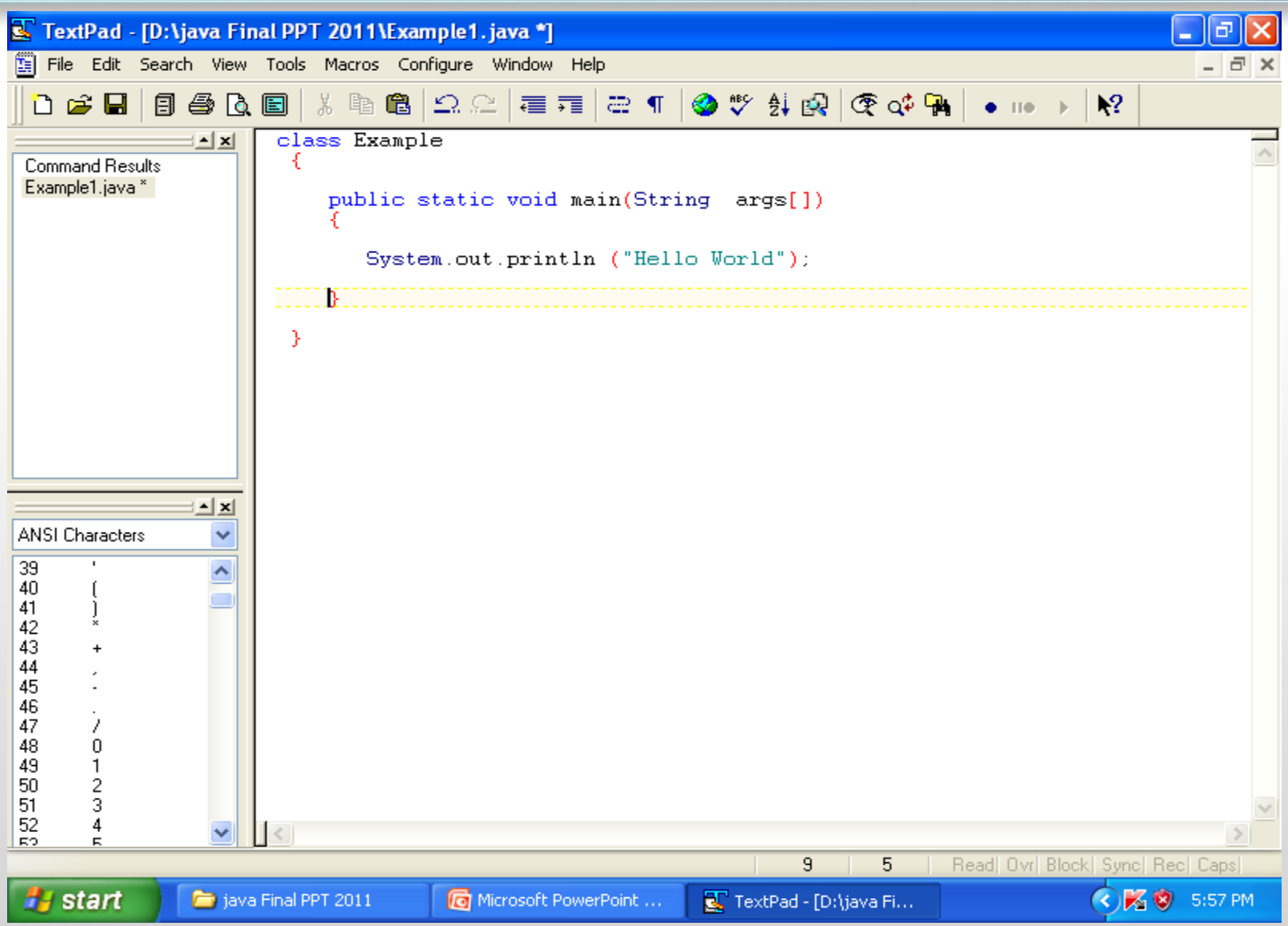
C:\ >jdk1.8\bin\ edit Example.java

C:\ >jdk1.8\bin\ javac Example.java

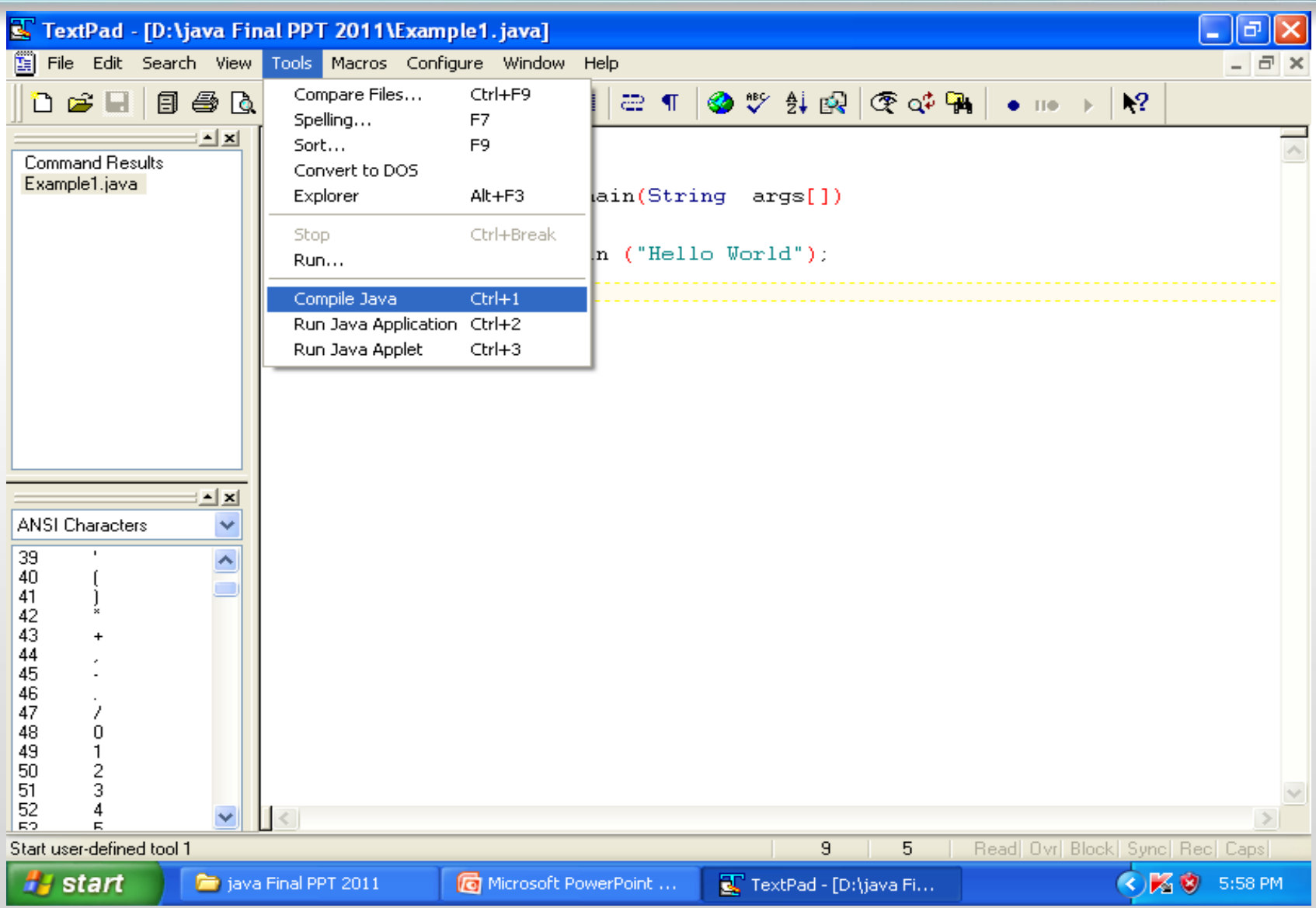
C:\ > jdk1.8\bin\ java Example

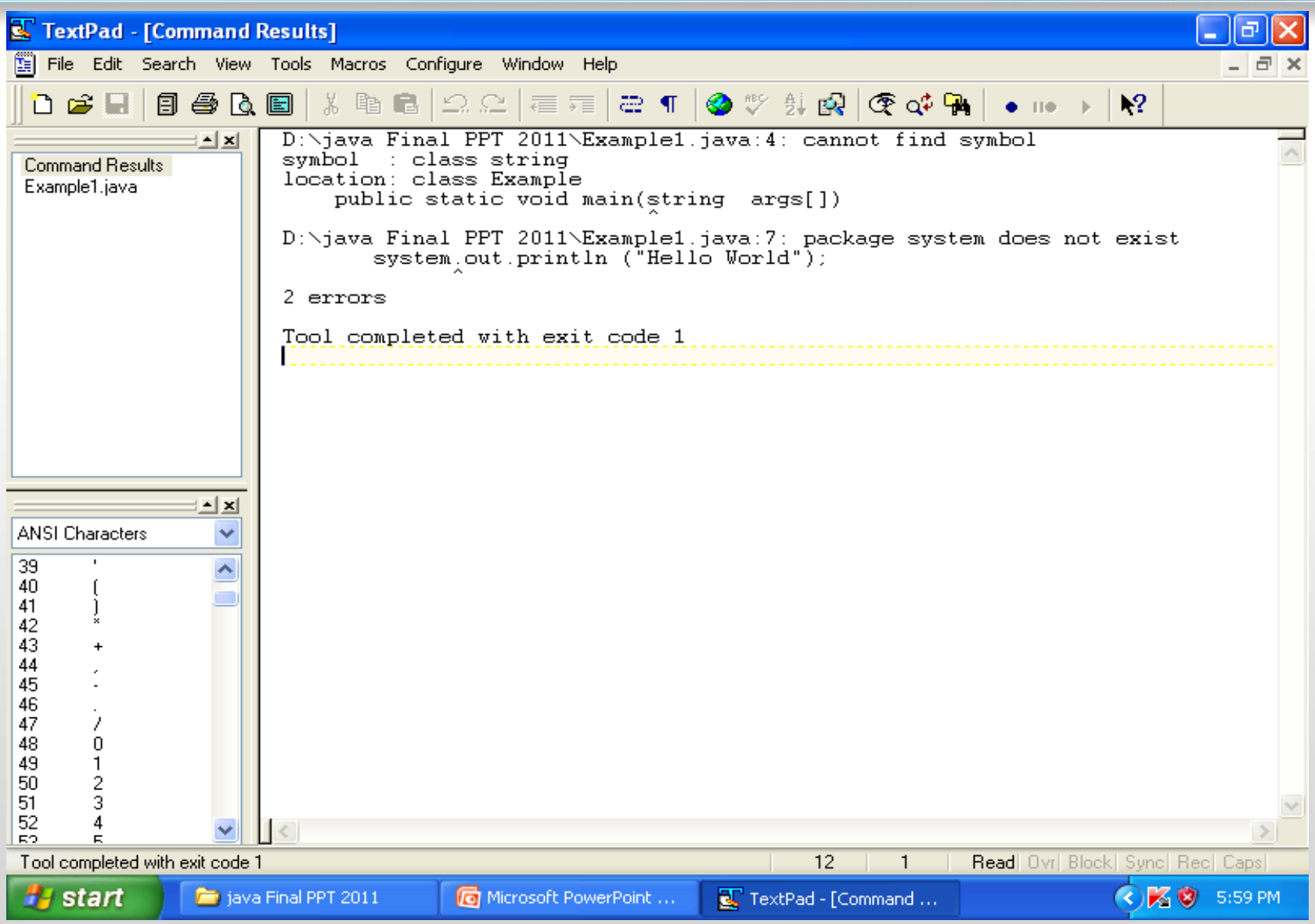
Hello World

Using TextPad

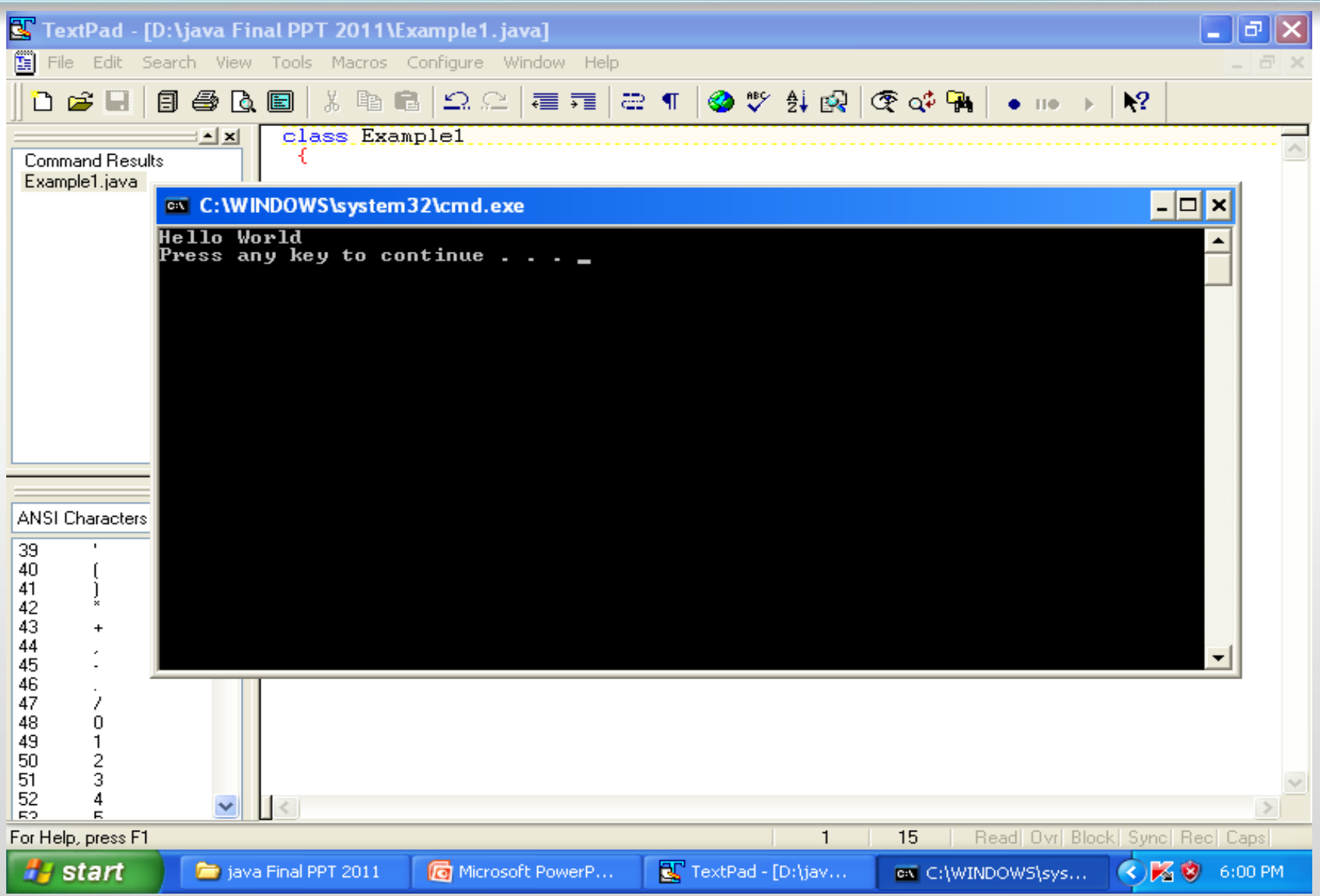


Compile Java – ctrl+1




TextPad - [Command Results]
File Edit Search View Tools Macros Configure Window Help
Command Results
Example1.java
D:\java Final PPT 2011\Example1.java:4: cannot find symbol
symbol : class string
location: class Example
public static void main(string args[])
D:\java Final PPT 2011\Example1.java:7: package system does not exist
system.out.println ("Hello World");
2 errors
Tool completed with exit code 1
ANSI Characters
39 '
40 (
41)
42 *
43 +
44 ,
45 -
46 .
47 /
48 0
49 1
50 2
51 3
52 4
Tool completed with exit code 1
12
1
Read Ovr Block Sync Rec Caps
start
java Final PPT 2011
Microsoft PowerPoint ...
TextPad - [Command ...
5:59 PM

Run Java Program- ctrl+2



Example2 : Variable Declaration

```
class Example2
{     public static void main(String args[])
     {          int  a,b,c;
                a=2;
                b=3;
                c = a+b;
                System.out.println(c);
                System.out.println("sum is"+c);
                System.out.println("sum of"+a+"and"+b+ "is"+c);
       }
}
```

OUTPUT : 5

sum is 5

Sum of 2 and 3 is 5

Example2 : Variable Declaration

```
class Example2
{     public static void main(String args[])
     {          int  a,b,c;
                a=2;
                b=3;
                c = a+b;
                System.out.println(c);
                System.out.println("sum is"+c);
                System.out.println("sum of"+a+"and"+b is"+c);
               System.out.println("sum of"+ (a+b));
       }
}
```

OUTPUT : 5

sum is 5

Primitive Data Types

- Integers: byte 8, short 16, int 32 and long 64
  - Java does not support unsigned integers

Name	Width in bits	Range	
long	64	-9,223,372,036,854,775,808	9,223,372,036,854,775,807
int	32	-2,147,483,648	2,147,483,647
short	16	-32,768	32,767
byte	8	-128	127

- Floating-Point: float 32, double 64

Name	Width in bits	Approximate Range	
double	64	4.9e-324	1.8e+308
float	32	1.4e-045	3.4e+038

- Character: char 16 (Unicode)
- Boolean: boolean 8 (true, false)

Reading input from keyboard

```
int a;
Scanner sc = new Scanner(System.in);
 a = sc.nextInt();
```

1) int nextInt()

It is used to read an integer value from the keyboard.

2) int nextFloat()

It is used to read a float value from the keyboard.

3) long nextLong()

It is used to read a long value from the keyboard.

4) String next()

It is used to read string value from the keyboard.

```
import java.util.Scanner;
```

```
import java.util.Scanner;
class prg3
{       public static void main(String args[])
        {       int a,b,c;
                Scanner sc = new Scanner(System.in);
                System.out.println("Enter a first number");
                a = sc.nextInt();
                System.out.println("Enter a second number");
                b = sc.nextInt();
                c = a+ b;
                System.out.println("sum is :"+c);
        }   }
```

```
import java.util.Scanner;
class GetInputFromUser{
  public static void main(String args[])  {
        int a;
        float b;
        String s;
        Scanner in = new Scanner(System.in);
        System.out.println("Enter a string");
        s = in.nextLine();
        System.out.println("You entered string "+s);
        System.out.println("Enter an integer");
        a = in.nextInt();
        System.out.println("You entered integer "+a);
        System.out.println("Enter a float");
        b = in.nextFloat();
        System.out.println("You entered float "+b);
  }
```

PresentationPoint

Data Types and Variables

- **Variables**
  - Syntax: type identifier [= value][, identifier [=value]…];
  - Examples:

```
int a, b, c;
int d = 3, e, f = 5;
byte z = 10;
double pi = 3.14159;
float k = 3.56f;
char x = 'x';
```

  - Types of variables
    - Local variables
    - Instance variables
    - Class / Static variables

Arithmetic Operators

Operator	Result
+	Addition
–	Subtraction (also unary minus)
*	Multiplication
/	Division
%	Modulus
++	Increment
+=	Addition assignment
–=	Subtraction assignment
*=	Multiplication assignment
/=	Division assignment
%=	Modulus assignment
– –	Decrement

Example

```
// Demonstrate the basic arithmetic operators.
class BasicMath
{       public static void main(String args[])
      {     System.out.println("Integer Arithmetic");
            int   a = 1 + 1;
            int   b = a * 3;
            int   c = b / 4;
            int   d = c - a;
            int   e = -d;
            System.out.println("a = " + a);
            System.out.println("b = " + b);
            System.out.println("c = " + c);
            System.out.println("d = " + d);
            System.out.println("e = " + e);
  } }
```

The modulus operator

```
// Demonstrate the % operator.
class Modulus
 {     public static void main(String args[])
      {
                int x = 42;
                double y = 42.25;
                System.out.println("x mod 10 = " + x % 10);
               System.out.println("y mod 10 = " + y % 10);
       }
}
```

output:

x mod 10 = 2

y mod 10 = 2.25

Relational Operator

Operator	Result
==	Equal to
!=	Not equal to
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to

Boolean Logical Operators

Operator	Result
&	Logical AND
|	Logical OR
^	Logical XOR (exclusive OR)

Decision Structures

```
if (condition)
    statement;
else if (condition)
    statement;
.
.
else
    statement;
```

Java program to illustrate if-else-if ladder

```
class ifelseifDemo
{
        public static void main(String args[])
        {
                int i = 20;

                if (i == 10)
                        System.out.println("i is 10");
                else if (i == 15)
                        System.out.println("i is 15");
                else if (i == 20)
                        System.out.println("i is 20");
                else
                        System.out.println("i is not present");
        }
}
```