

OBJECT ORIENTED PROGRAMMING

III Semester

CCE A Section

Introduction

The birth of modern programming language C, the need for C++, Java, Importance of Java in the internet, security, portability, the byte code. An overview of Java, OOP, Two paradigms, abstraction, the three OOP Principles.

Data Types, Variable and Arrays

Simple types, integers, floating point types, characters, Booleans, variables – declaring variable, dynamic initialization, the scope and life time of variables, type conversion and casting, arrays-one dimensional arrays and multi-dimensional arrays.

Operators and control statements

Arithmetic operators, bitwise operators, relational operators, logical operators, assignment operators, ternary operators, operator precedence. Control statements – if, switch, while, do-while, for nested loops, break, continue.

Classes and Inheritance

Class fundamentals, declaring objects, assigning object references variables, introducing methods, constructors, overloading method, using objects as parameters, argument passing, returning objects, recursion, use of static and final key word, nested and inner class, using command line arguments. Inheritance – basics, using super, creating a multi-level hierarchy, when constructor are called, method overriding, dynamic method dispatch, using abstract classes, using final with inheritance, Wrapper classes.

Packages and Interfaces

Packages, defining a package, use of CLASSPATH, package example, access protection, importing packages, Interfaces – defining an interface, implementing interfaces, applying interfaces, variables in interfaces, extending interfaces.

Array list and Vectors

The list interface, Array list class, obtaining an array from an array list, Vector.

String Handling

String constructors, string length, special operations, character extraction, string comparison, searching strings, modifying a string, string buffer, string tokenizer.

Exception Handling

Fundamentals, exception types, uncaught exception, using try and catch, multiple catch clauses, nested try statements, throw, throws, finally, Java's built in exception, creating exception subclasses, using exception.

Thread Concepts

The java thread model, thread priorities, synchronization, thread class and runnable interface, the main thread, creating a thread, creating multiple threads, using is Alive and join, inter thread communication.

Input/Output

Java I/O classes and Interfaces, File – directories, using filename filter, the stream classes, the byte streams-input stream, output stream, file input stream, file output stream, byte array input stream, byte array output stream, random access files. The character streams- Reader, Writer, FileReader, FileWriter, BufferedReader, BufferedWriter. Serialization, Serializable, Object Output, Object Output Stream, Object Input, Object Input Stream.

Swings

Swing fundamentals. Writing swing application, Swing library, layouts and controls. Introduction to event handling

References :

Herbert Schildt, *Java The Complete Reference (9e)*, Tata McGrawHill 2014.

Cay S. Horstmann & Gary Cornell, *Core Java Volume I – Fundamentals (9e)*, Prentice Hall 2013.

Cay S. Horstmann & Gary Cornell, *Core Java Volume II – Advanced Features (9e)*, Prentice Hall 2013.

Course Outcome :

- CO1 : Develop simple applications using JAVA. (data types, operators, arrays)
- CO2 : Understand how object oriented concepts can be implemented using JAVA
- CO3 : Use inbuilt library packages of JAVA
- CO4 : Develop java applications using object oriented concepts.
- CO5 : Write simple concurrent programs using threads.

JAVA

- Java was designed by Sun Microsystems.
- First release in January 1996.
- Developed by James Gosling.
- For controlling simple home devices with embedding microcomputers.
- Used to create desk top applications and web applications.

Java Features

- Simple
- Secure
- Portable
- Object-oriented
- Robust
- Multithreaded
- Architecture-neutral
- Interpreted
- Distributed
- Dynamic

Simple

- Looks familiar to existing programmers: related to C and C++
- Omits many rarely used, poorly understood, confusing features of C++, like operator overloading, multiple inheritance etc.
- Contains no *goto* statement, but *break* and *continue*
- has no pointers
- A rich predefined class library

Secure

- No memory pointers
- Programs runs inside the virtual machine sandbox.
- Array index limit checking
- Automatic memory management -GC

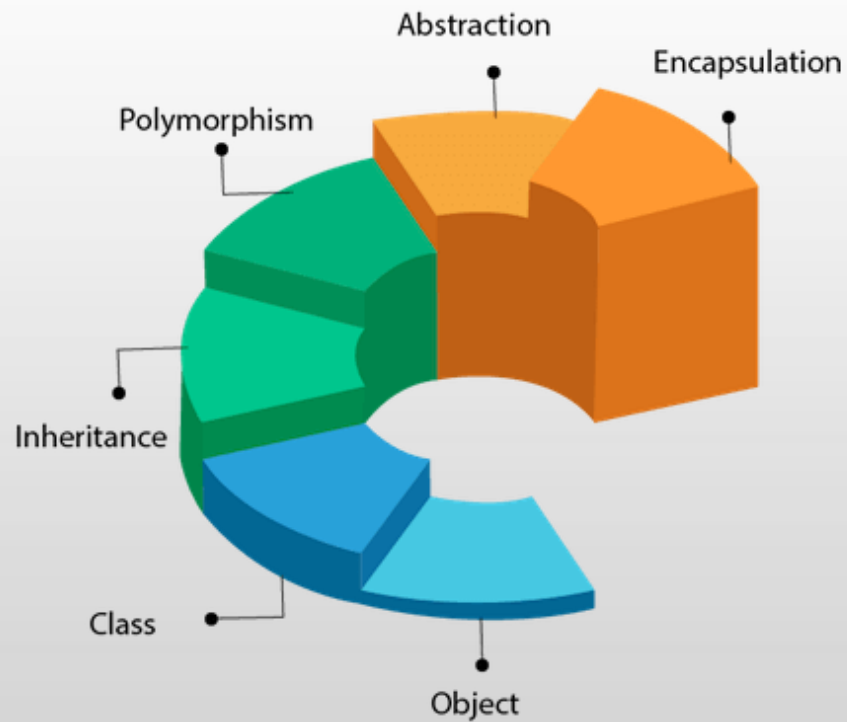
Portable

- Java compiler generates bytecodes
- Runtime systems for various platforms
- Size and behavior of basic data types defined
- Write once, run anywhere

Object Oriented

- Focus on the *data* in our application and *methods* that manipulate that data.
- *class* is a collection of data and methods that operate on that data.
- The data and methods describe the state and behavior of an *object*.
- Classes are arranged in a hierarchy, a subclass can inherit behavior from its superclass.
- Has extensive set of classes, arranged in *packages*.

OOPs (Object-Oriented Programming System)



Robust

- Memory Management
- Exception Handling

Multithreaded

- Supports multithreaded programming, which allows to write programs that do many things simultaneously

Architectural neutral

- Write once- run anywhere

Interpreted

- Enables the creation of cross-platform programs by compiling into an intermediate representation called Java bytecode .
- This code can be interpreted on any system that provides a Java Virtual Machine.

Distributed

- Java has extensive TCP/IP networking capabilities.
- Has a feature called Remote Method Invocation (RMI).
- Library routines for HTTP, FTP exist.
- Accessing a resource using a URL is not much different from accessing a file.

Java and C

Java does not include keywords goto, sizeof and typedef.

Java does not contain the datatypes struct, union and enum.

Java does not define the type modifiers auto, extern, register, signed and unsigned.

Java does not support explicit pointer types.

Java does not have a preprocessor and therefore we can not use #define, #include.

Java adds new operator such as **instanceof** and **>>>**

Java adds **labeled break and continue** statements

Java and C++

Java adds many features required for object oriented programming

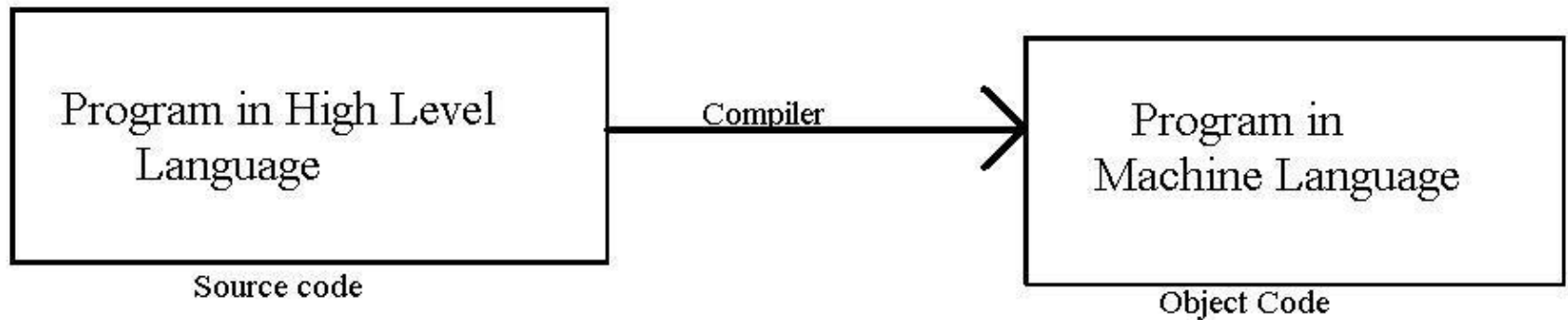
Java does not support operator overloading.

Java does not support multiple inheritance. Has a new feature called “interface”.

Java has replaced the destructor function with a finalize function.

Java does not use pointers.

Compilation Process



Program Life Cycle

Compilation & Execution:

Resides in
secondary memory

Program

On compilation

Byte Code

Resides in
secondary memory

On execution

Byte Code

Loaded to main
memory (process)

On execution

Output

Generated on
output devices

Input devices

Interacts with

Java Programming Language

- Object Orientation
- Programmer friendly features
- Robust
- Platform independent

Java Byte Code

- Intermediate code generated by the Java compiler
- Also called Java class file
- Facilitates "Write Once, Run Anywhere"

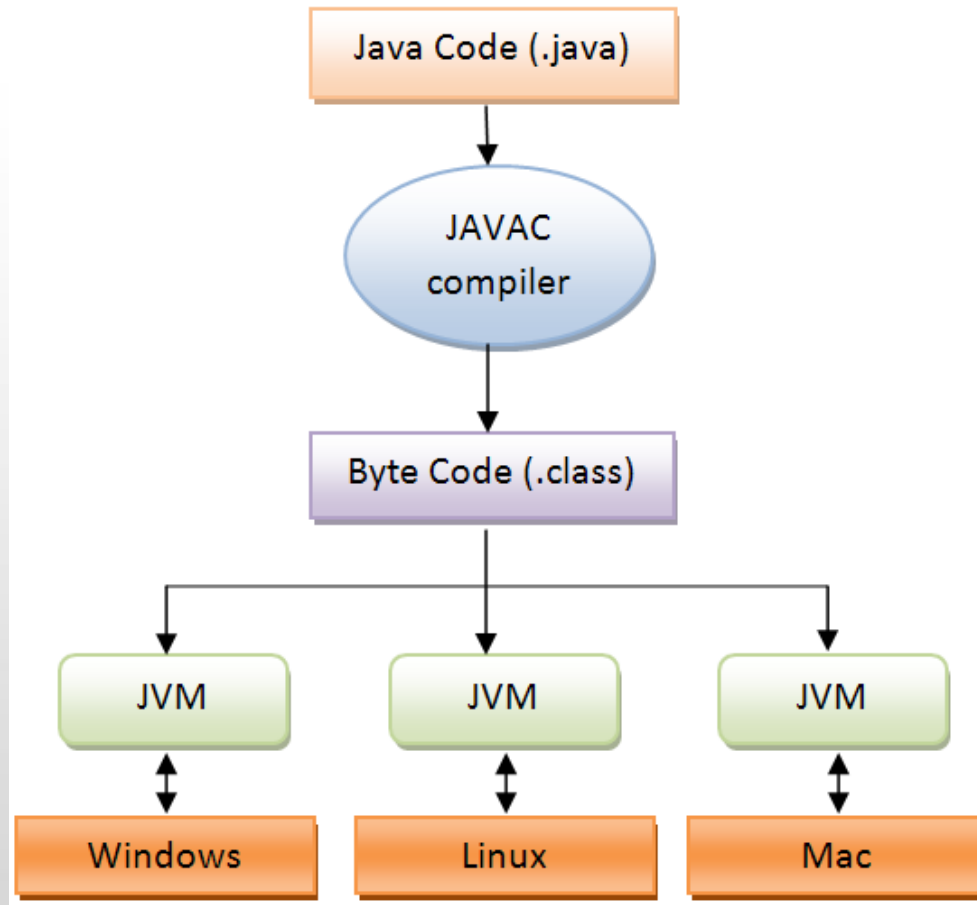
Java Virtual Machine

- Loads byte code and executes them
- It is platform dependent
- Different for different platforms

Java Virtual Machine (JVM)

- The Java virtual machine (JVM) is a software implementation of a computer.
- JVM executes programs like a real machine.
- The JVM is written specifically for a specific operating system.
 - e.g. for Linux a special implementation is required as well as for Windows.
- Java programs are compiled by the Java compiler into *bytecode*.
- The JVM interprets this **bytecode** and executes the Java program.

Java Virtual Machine (JVM)



Java Development Kit (JDK)

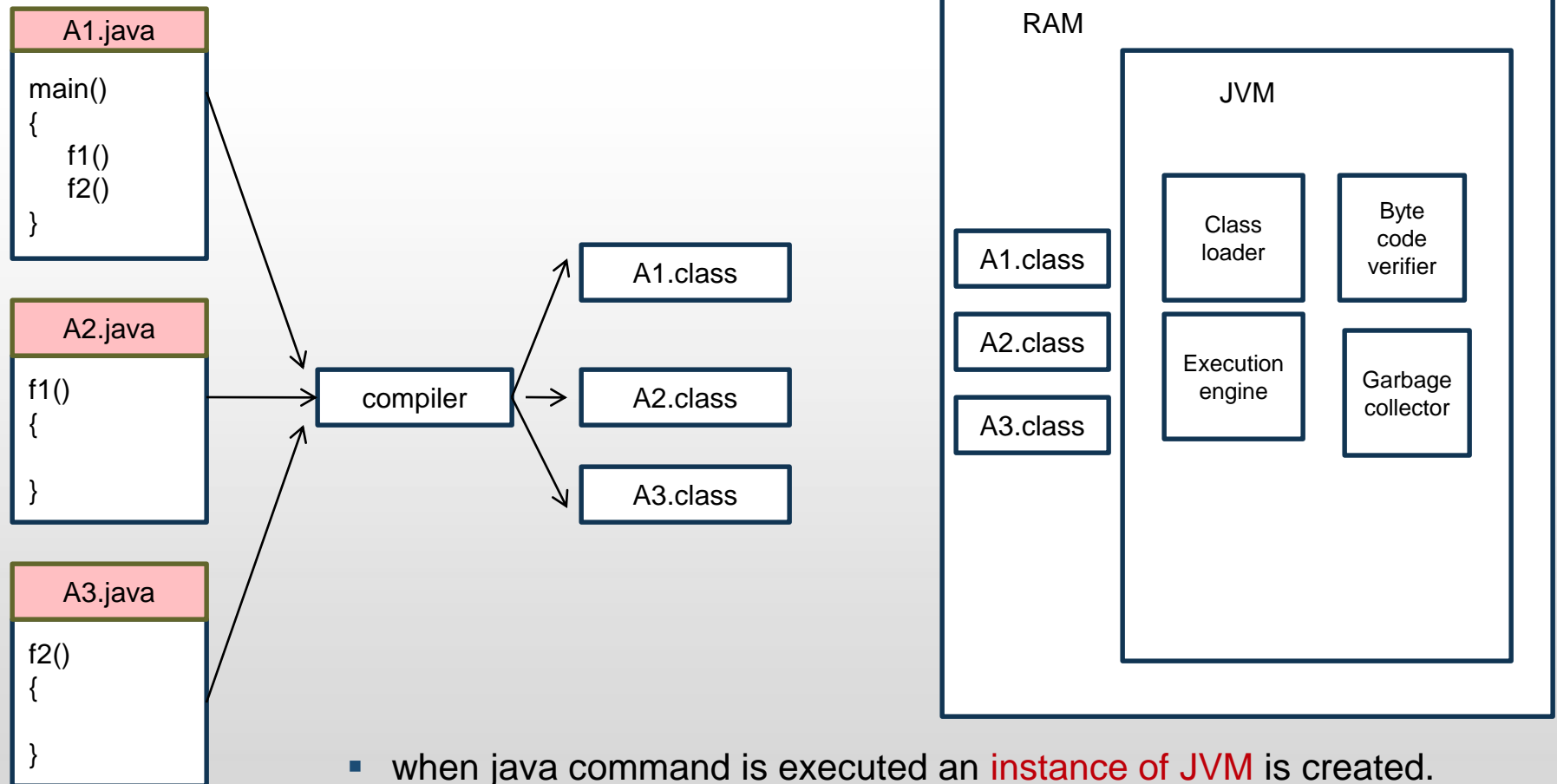
Java Runtime Environment (JRE)

Java Virtual Machine (JVM)

$\text{Jdk} = \text{jre} + \text{development tools}$

$\text{Jre} = \text{jvm} + \text{library classes}$

Java Language



- when java command is executed an **instance of JVM** is created.
- The class loader is responsible for bringing the class file into the JVM

Example 1

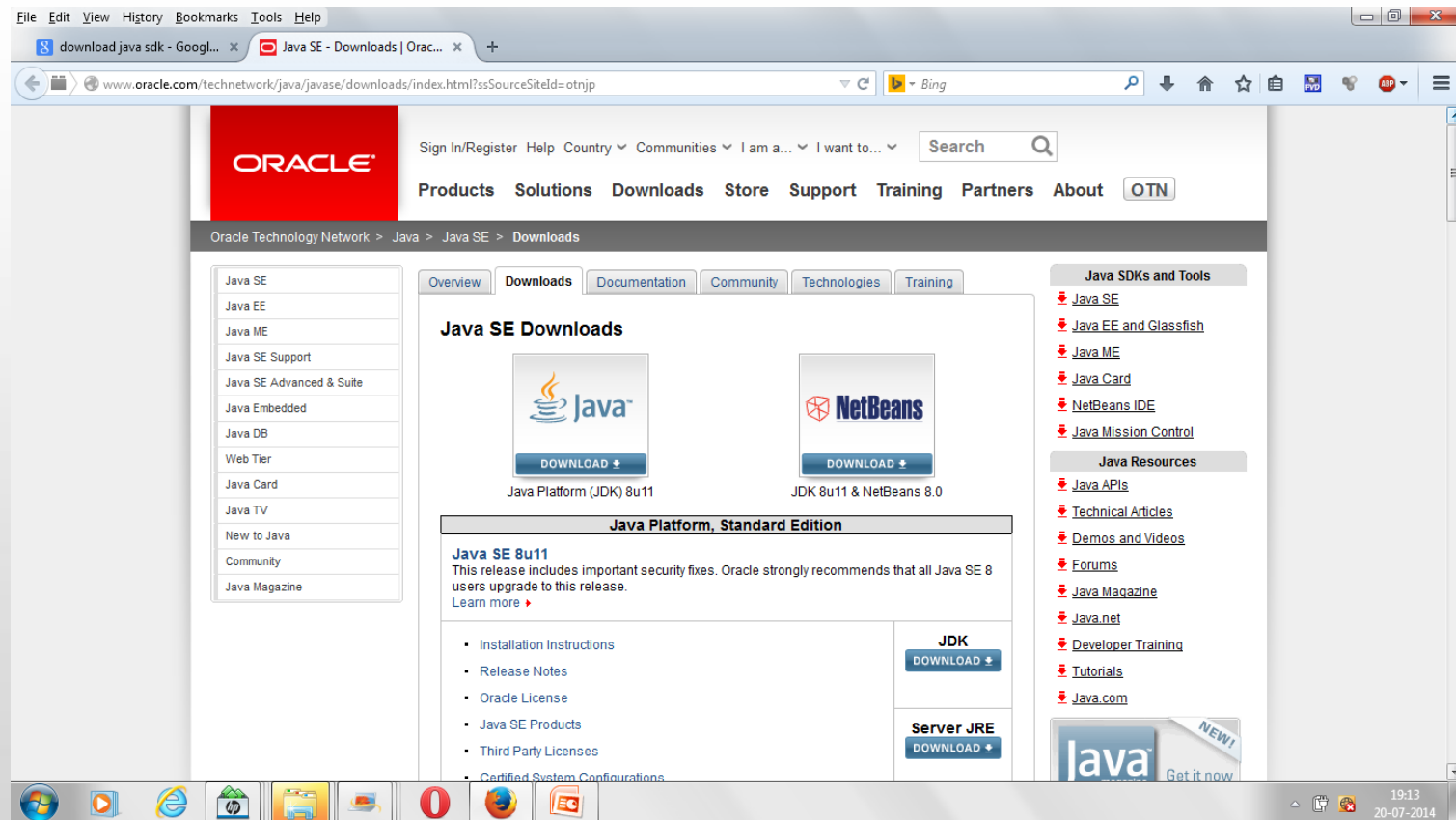
```
class Example
{
    public static void main(String args[] )
    {
        System.out.println ("Hello World");
    }
}
```

Note : String, System

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

- main must be declared as **public** since it must be called by code outside of class when the program is started.
- The keyword **static** allows main() to be called without having to instantiating a particular instance of the class
- **main** is called by the Java Virtual Machine before any objects are made.

Downloading and installing JDK



Downloading and installing JDK



 Menu



 Sign In ▾

 Country/Region ▾

 Call

Oracle Technology Network / Java / Java SE / Downloads

Java SE

Java EE

Java ME

Java SE Subscription

Java Embedded

Java Card

Java TV

Community

Java Magazine

Overview

Downloads

Documentation

Community

Technologies

Training

Java SE Downloads



DOWNLOAD ▾

Java Platform (JDK) 10

Java Platform, Standard Edition

Java SE 10.0.2
Java SE 10.0.2 is the latest feature release for the Java SE Platform
[Learn more ▶](#)

- Installation Instructions
- Release Notes
- Oracle License
- Java SE Licensing Information User Manual
 - Includes Third Party Licenses
- Certified System Configurations
- Readme

JDK

DOWNLOAD ▾

Server JRE

DOWNLOAD ▾

JRE

DOWNLOAD ▾

Java SDKs and Tools

- [Java SE](#)
- [Java EE and Glassfish](#)
- [Java ME](#)
- [Java Card](#)
- [NetBeans IDE](#)
- [Java Mission Control](#)

Java Resources

- [Java APIs](#)
- [Technical Articles](#)
- [Demos and Videos](#)
- [Forums](#)
- [Java Magazine](#)
- [Developer Training](#)
- [Tutorials](#)
- [Java.com](#)

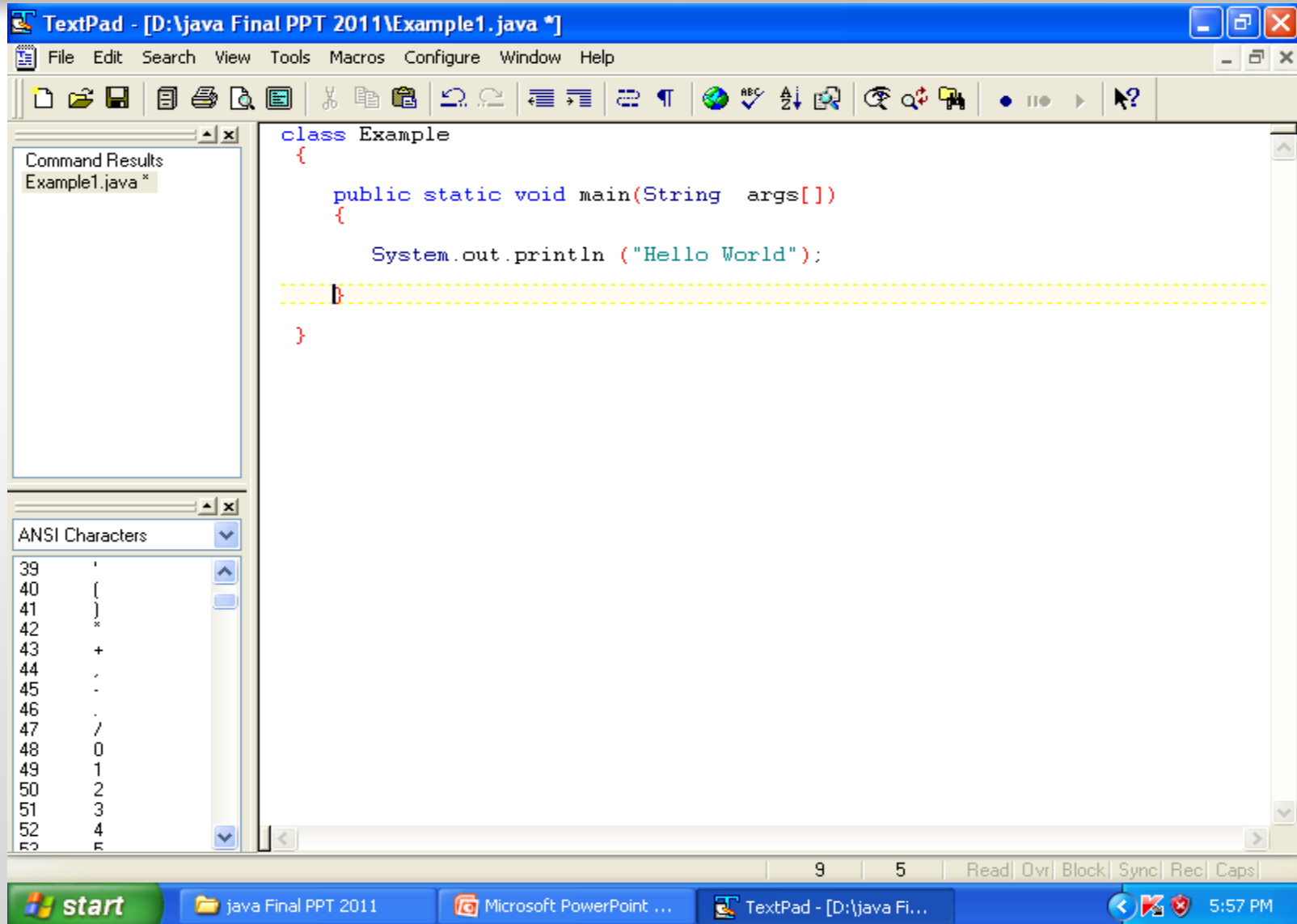
Compiling & Executing Java Program

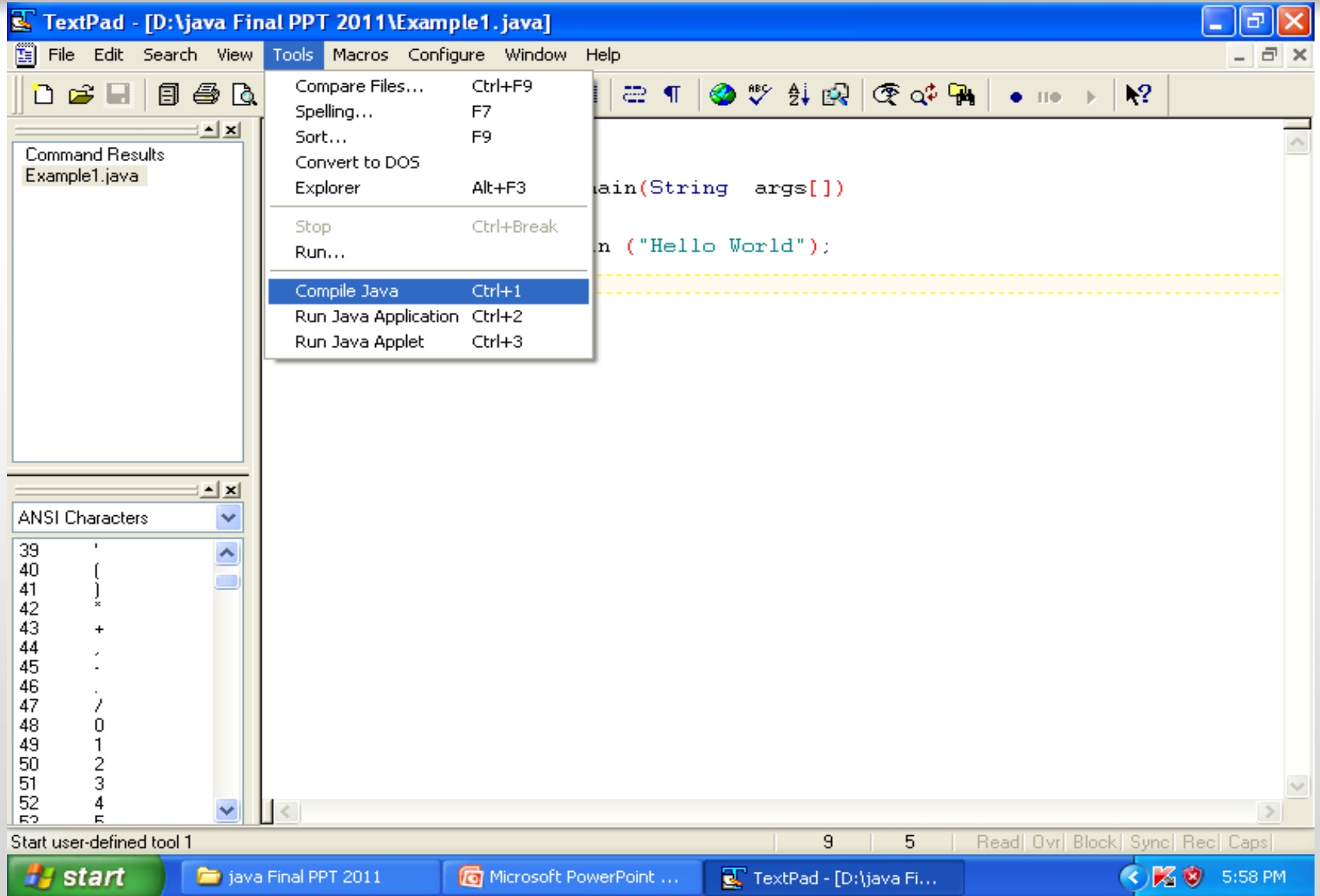
```
C:\>jdk1.8\bin\    edit Example.java
```

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

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

```
Hello World
```



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
```

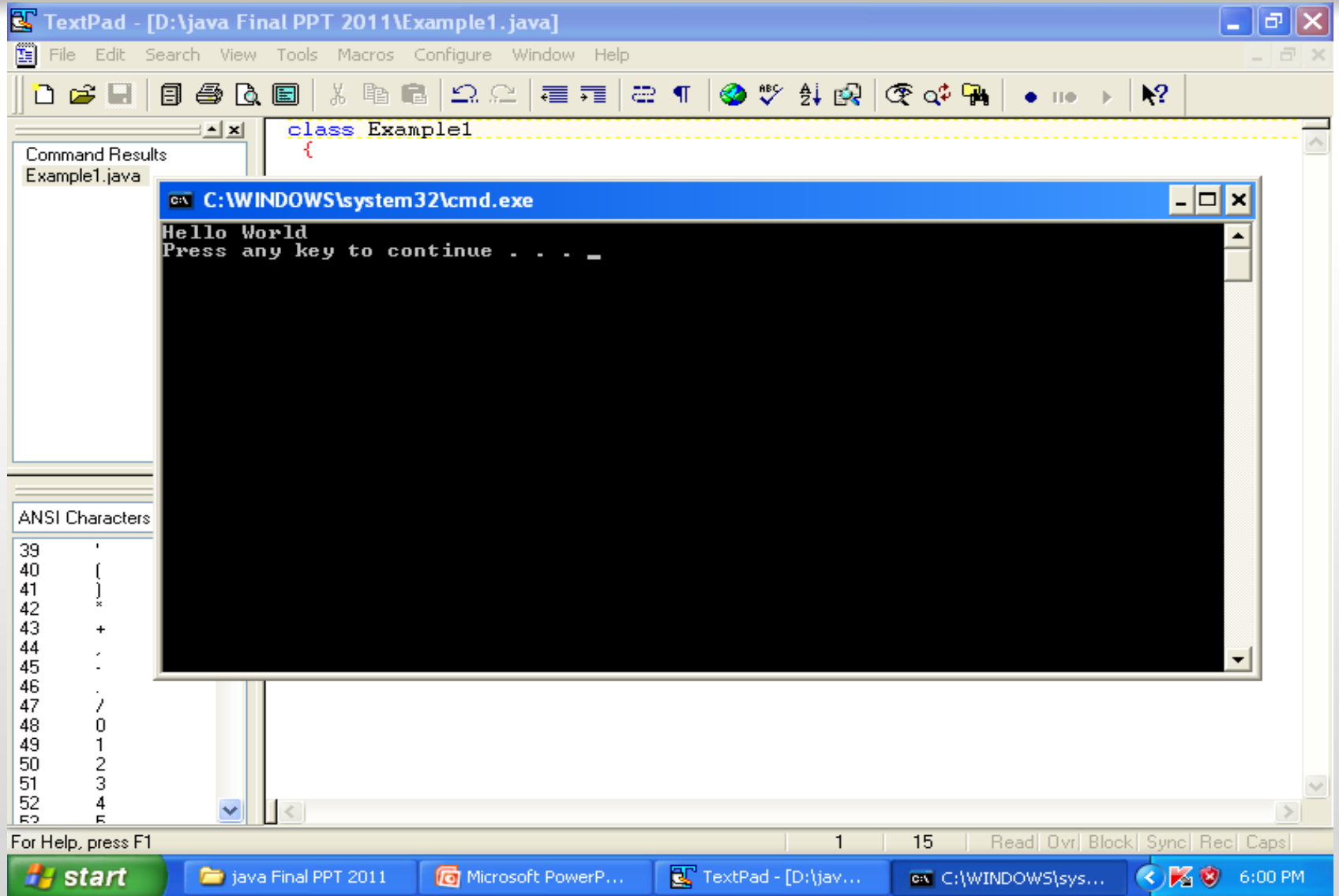
ANSI Characters

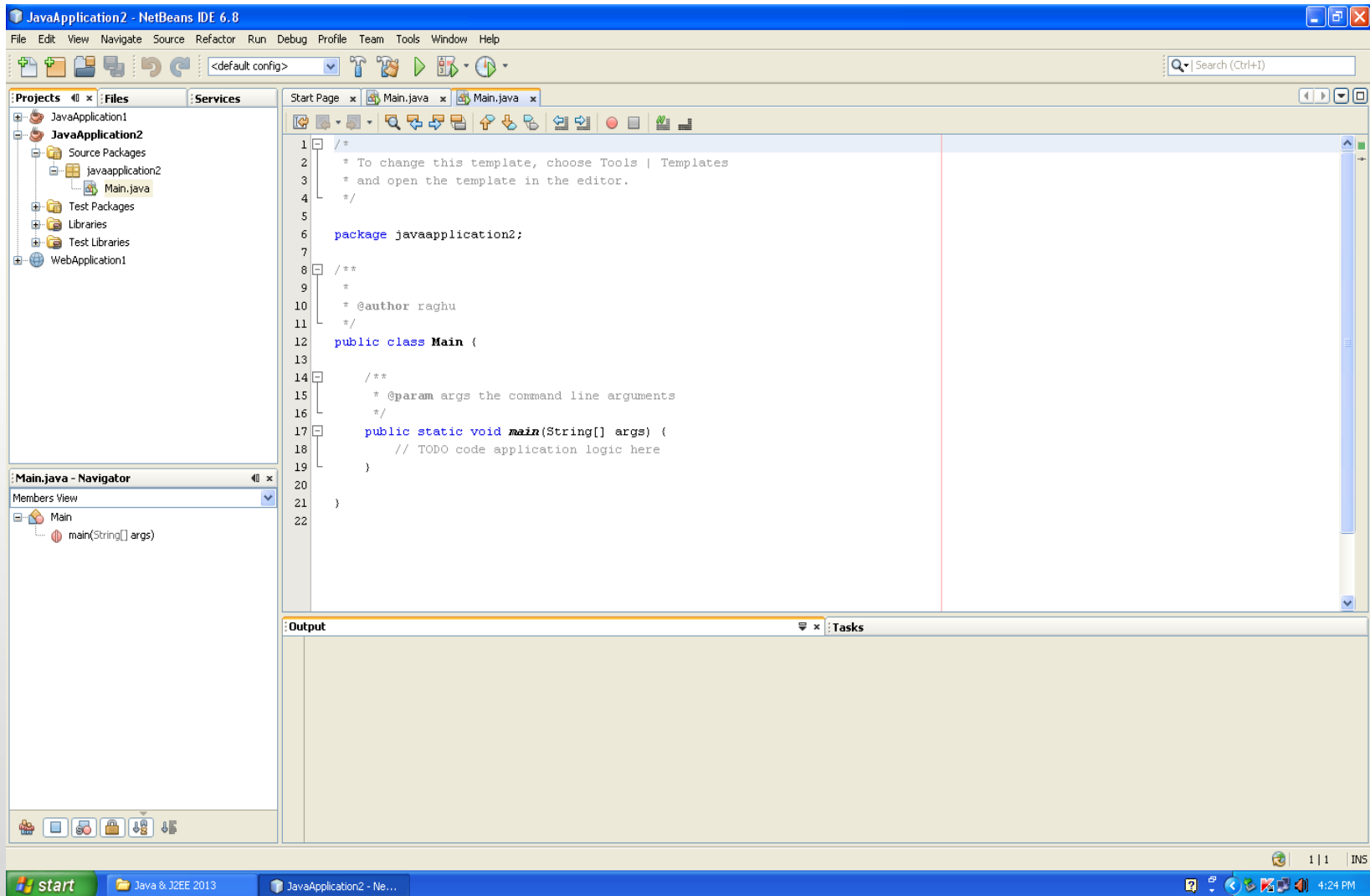
39	'
40	{
41	}
42	*
43	+
44	.
45	-
46	.
47	/
48	0
49	1
50	2
51	3
52	4
53	5

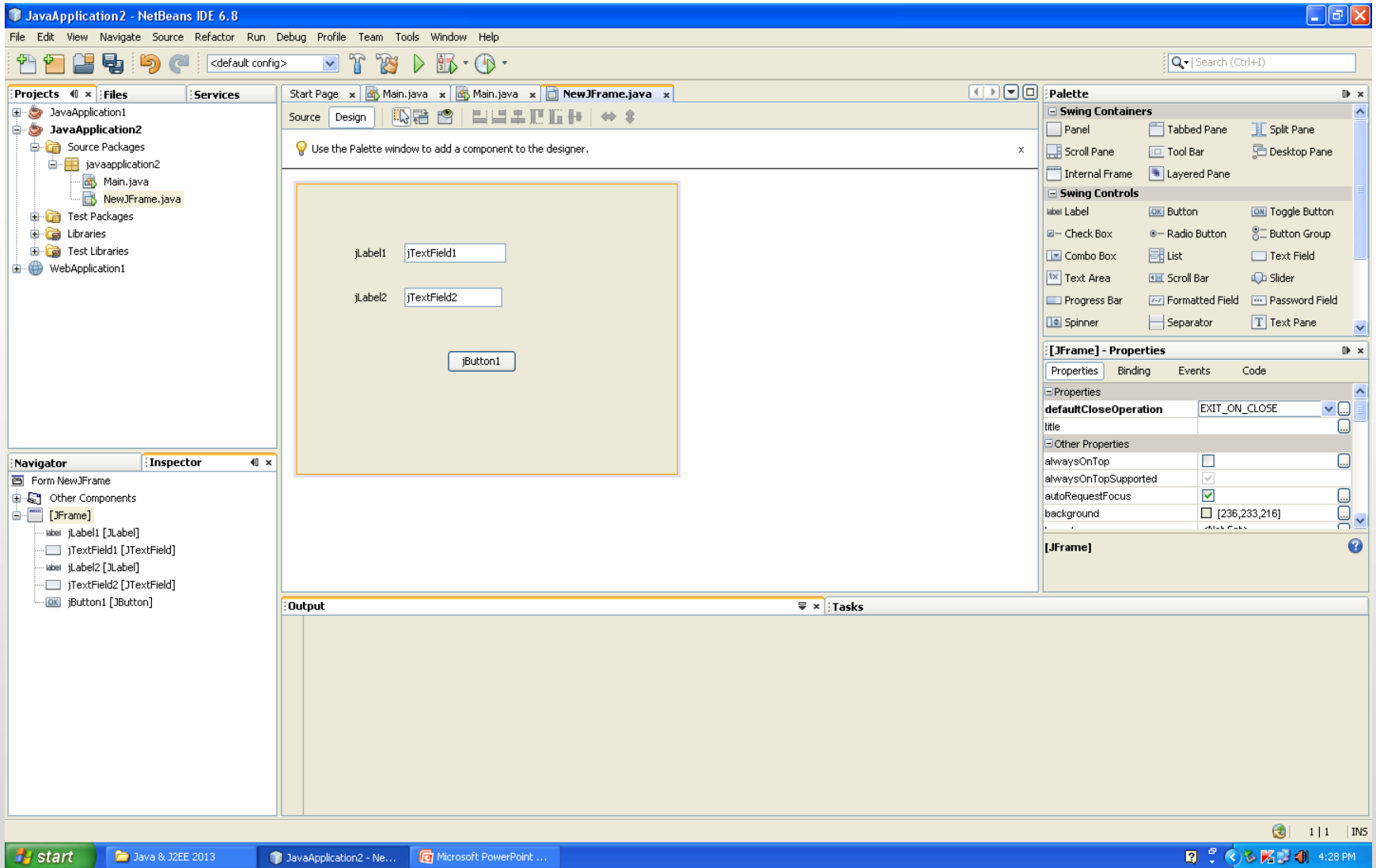
Tool completed with exit code 1

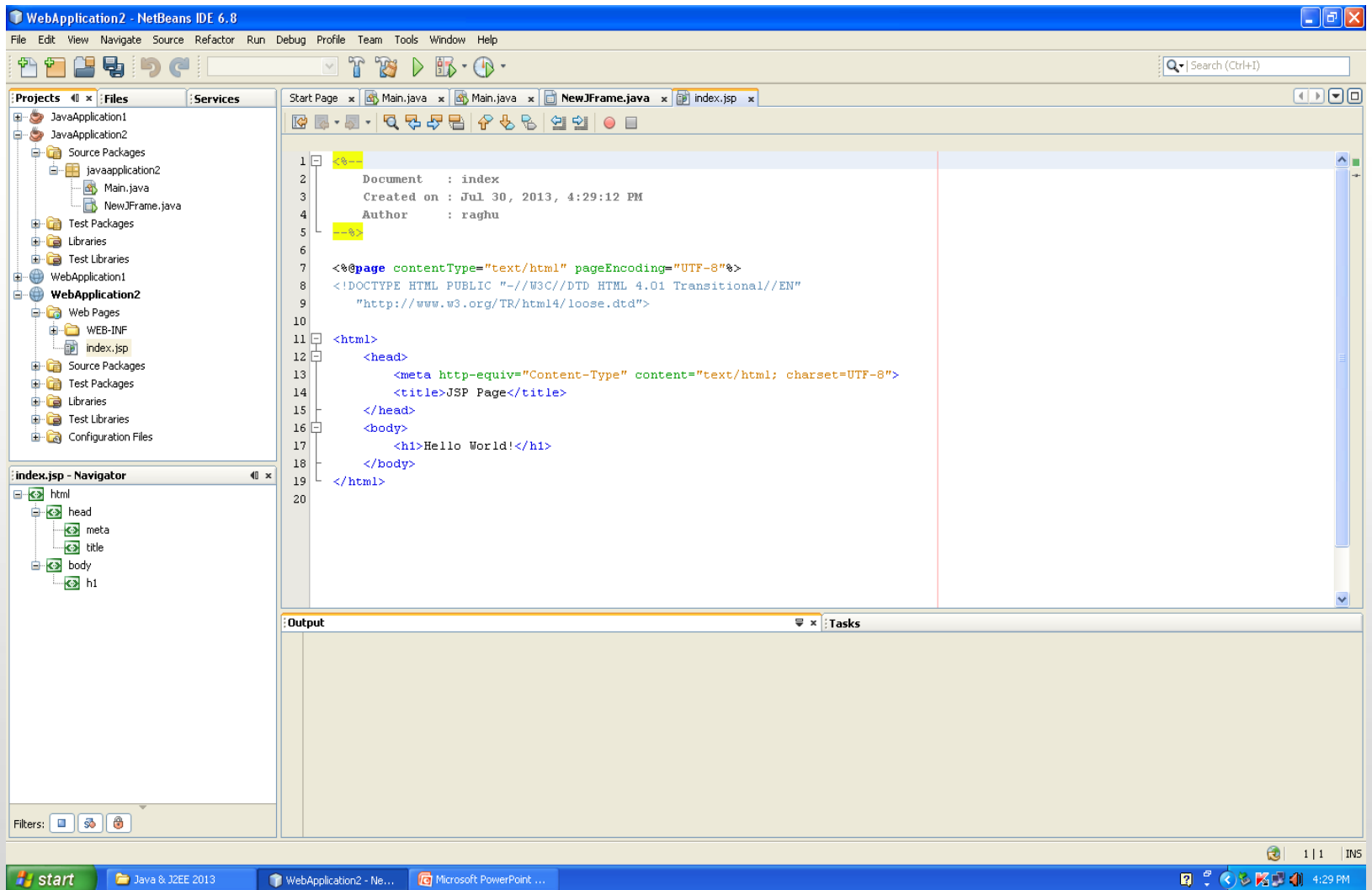
12 1 Read Ovr Block Sync Rec Caps

start java Final PPT 2011 Microsoft PowerPoint ... TextPad - [Command ... 5:59 PM









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