

Object Oriented Programming in JAVA

Introduction

- Welcome to the course **Object Oriented Programming in JAVA**. This course will cover a core set of computer science concepts needed to create a modern software application using Java.

Course Objectives

On completion of this course we will be able to:

1. Identify the **importance** of Java .
2. Identify the additional **features** of Java compared to **C++** .
3. Identify the difference between **Compiler** and **Interpreter** .
4. Identify the difference between **applet** and **application** .
5. Apply **Object Oriented Principles** of **Encapsulations**, **Data abstraction**, **Inheritance**, **Polymorphism**.
6. Program using **java API** (Application Programming Interface).
7. Program using **Exception Handling**, **Files** and **Threads** .
8. Program Using **applets** and **swings** .

Course Syllabus

UNIT	CONCEPTS to be covered
UNIT-I	JAVA Basics
UNIT-II	Inheritance
UNIT-III	Data structures creation and manipulation in java
UNIT-IV	Exception Handling
UNIT-V	GUI Programming With JAVA

JAVA Basics

Why **Java** is Important

- **Two reasons** :
 - Trouble with **C/C++** language is that they are not portable and are not platform independent languages.
 - Emergence of World Wide Web, which demanded portable programs
- **Portability** and **security** necessitated the invention of Java

History

- James Gosling - Sun Microsystems
- Co founder – Vinod Khosla
- Oak - Java, May 20, 1995, Sun World
- JDK Evolutions
 - JDK 1.0 (January 23, 1996)
 - JDK 1.1 (February 19, 1997)
 - J2SE 1.2 (December 8, 1998)
 - J2SE 1.3 (May 8, 2000)
 - J2SE 1.4 (February 6, 2002)
 - J2SE 5.0 (September 30, 2004)
 - Java SE 6 (December 11, 2006)
 - Java SE 7 (July 28, 2011)

Cont..

- Java Editions.

- **J2SE**(Java 2 Standard Edition) - to develop client-side standalone applications or applets.
- **J2ME**(Java 2 Micro Edition) - to develop applications for mobile devices such as cell phones.
- **J2EE**(Java 2 Enterprise Edition) - to develop server-side applications such as Java servlets and Java ServerPages.

What is java?

- A general-purpose **object-oriented language**.
- **W**rite **O**nce **R**un **A**n anywhere (WORA).
- Designed for easy **Web/Internet** applications.
- **Widespread** acceptance.

How is Java different from C...

- **C Language:**

- Major difference is that C is a **structure oriented language** and Java is an **object oriented language** and has mechanism to define classes and objects.
- Java does not support an explicit **pointer** type
- Java does not have **preprocessor**, so we cant use #define, #include and #ifdef statements.
- Java does not include structures, unions and enum data types.
- Java does not include keywords like goto, sizeof and typedef.
- Java adds labeled break and continue statements.
- Java adds many features required for object oriented programming.

How is **Java** different from **C++**...

- C++ language

Features removed in java:

- Java doesn't support **pointers** to avoid **unauthorized** access of **memory locations**.
- Java does not include structures, unions and enum data types.
- Java does not support **operator over loading**.
- Preprocessor plays less important role in C++ and so **eliminated** entirely in java.
- Java does not perform **automatic** type conversions that result in loss of **precision**.

Cont...

- Java does not support **global variables**. Every method and variable is declared within a **class** and forms part of that class.
- Java does not allow **default arguments**.
- Java does not support inheritance of **multiple** super classes by a sub class (i.e., **multiple inheritance**). This is accomplished by using '**interface**' concept.
- It is not possible to declare **unsigned integers** in java.
- In java objects are passed by **reference** only. In C++ objects may be passed by **value** or **reference**.

Cont ...

New features added in Java:

- **Multithreading**, that allows two or more pieces of the same program to execute concurrently.
- C++ has a set of library functions that use a common header file. But java replaces it with its own set of **API classes**.
- It adds **packages** and **interfaces**.
- Java supports automatic **garbage collection**.
- **break** and **continue** statements have been enhanced in java to accept labels as targets.
- The use of **unicode** characters ensures portability.

Cont ...

Features that differ:

- Though **C++** and **java** supports Boolean data type, C++ takes any **nonzero value** as true and **zero as** false. True and false in java are predefined literals that are values for a boolean expression.
- Java has replaced the **destructor** function with a **finalize()** function.
- C++ supports exception handling that is similar to java's. However, in C++ there is no requirement that a thrown exception be caught.

Characteristics of Java

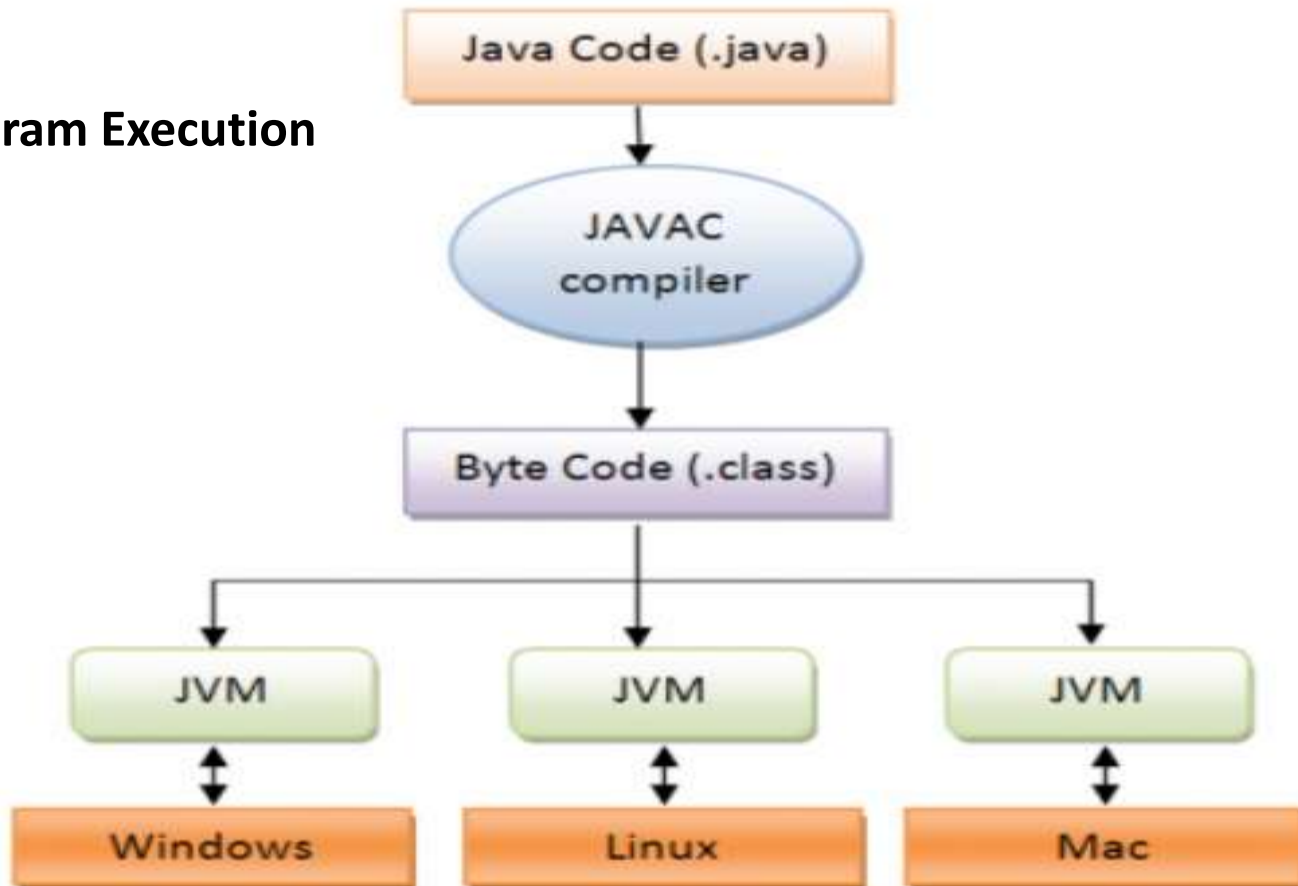
- Java is simple
- Java is object-oriented
- Java is distributed
- Java is interpreted
- Java is robust
- Java is architecture-neutral
- Java is portable
- Java's performance
- Java is multithreaded
- Java is dynamic
- Java is secure

Java Environment

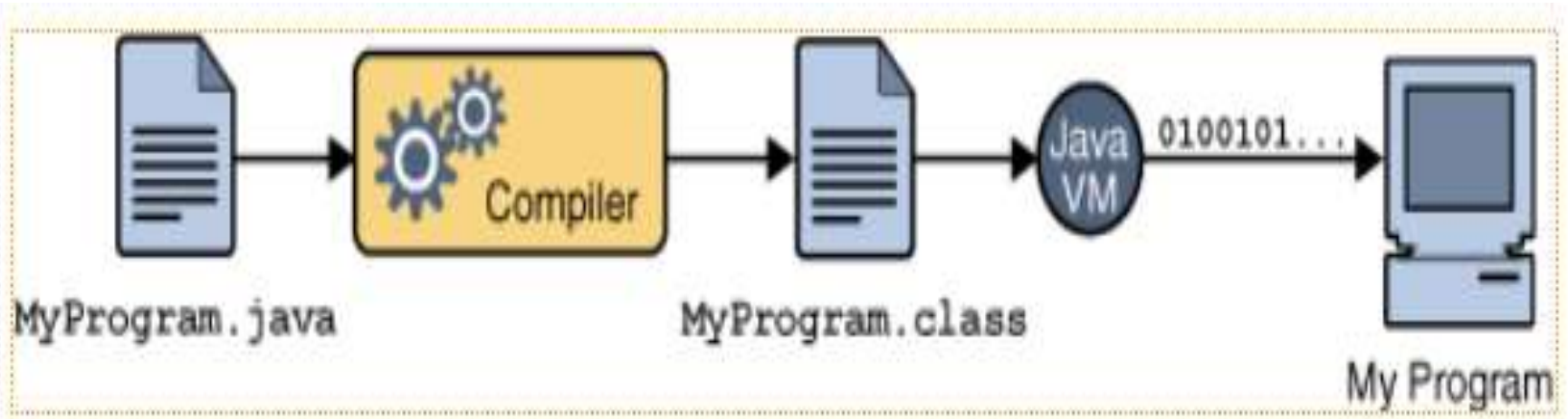
- Java includes many development tools, classes and methods
 - Development tools are part of Java Development Kit (JDK) and
 - The classes and methods are part of **Java Standard Library** (JSL), also known as **Application Programming Interface** (API).
- JDK constitutes of tools like **java compiler**, java interpreter and many.
- **API** includes hundreds of **classes** and **methods** grouped into several **packages** according to their functionality.

Java is architecture-neutral

JAVA Program Execution



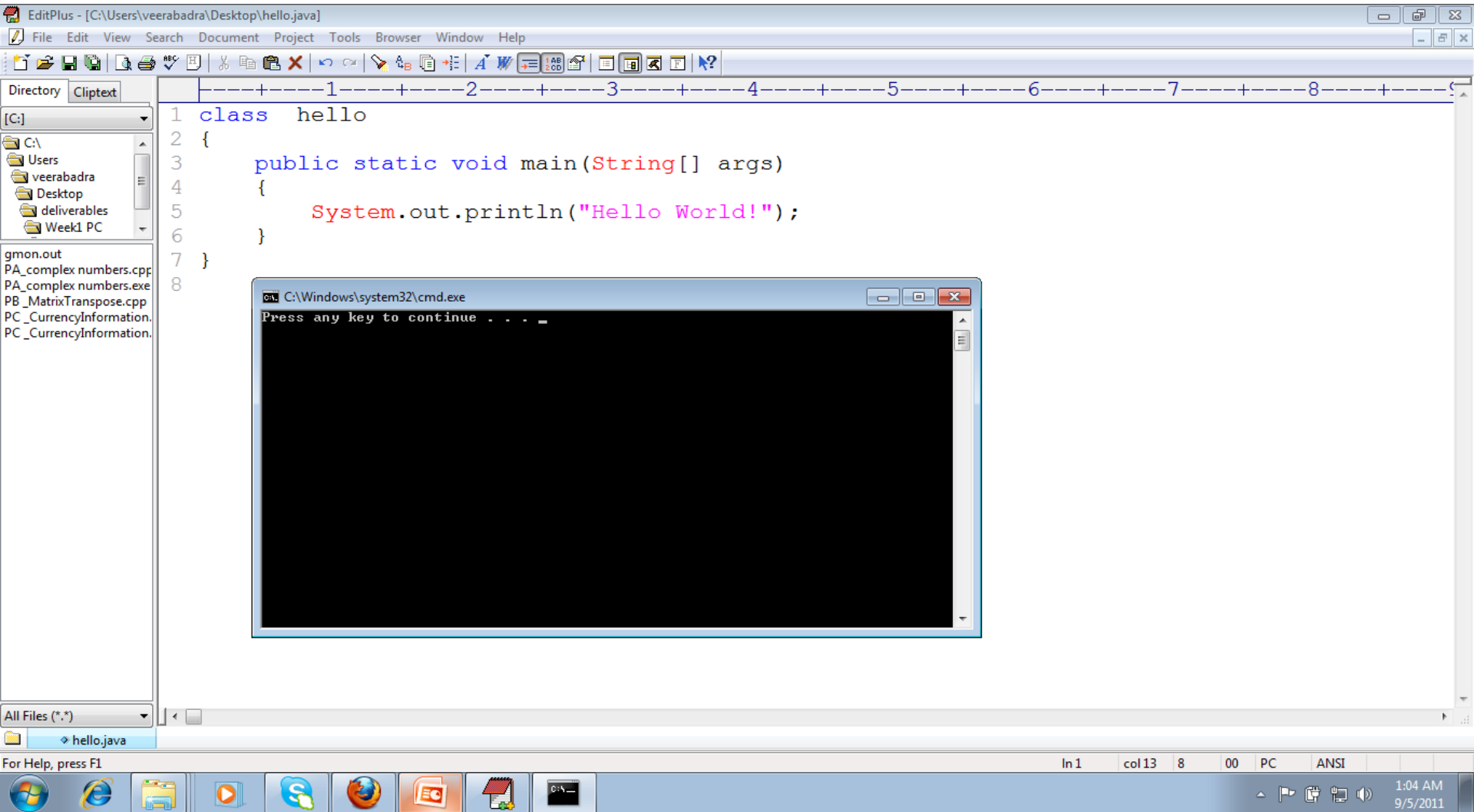
WORA(Write Once Run Anywhere)



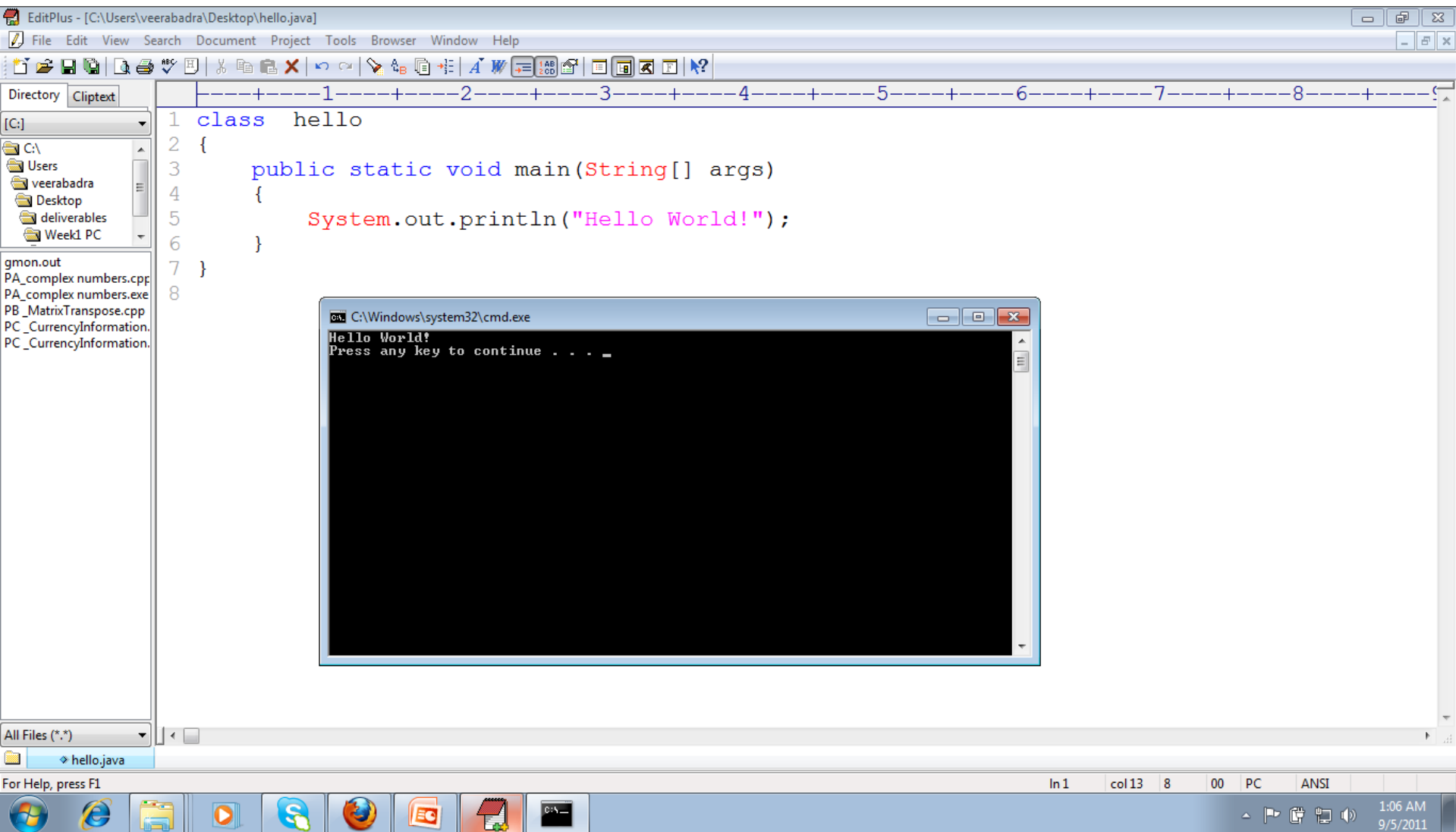
Editplus for Java Programming

- **Edit Plus Software:**
- **EditPlus** is a 32-bit text editor for the **Microsoft Windows** operating system.
- The editor contains tools for **programmers**, including **syntax highlighting** (and support for custom syntax files), file type conversions, line ending conversion (between **Linux**, **Windows** and **Mac** styles), regular expressions for search-and-replace, spell check etc).

Hello world Program in java



Execution of Hello world Program



The screenshot displays an IDE window titled "EditPlus - [C:\Users\veerabdra\Desktop\hello.java]". The main editor area contains the following Java code:

```
1 class hello
2 {
3     public static void main(String[] args)
4     {
5         System.out.println("Hello World!");
6     }
7 }
8
```

On the left side, there is a "Directory" pane showing the file structure of the project, including folders like "Users", "veerabdra", "Desktop", "deliverables", and "Week1 PC". Below the directory pane, a list of files is visible, including "gmon.out", "PA_complex numbers.cpp", "PA_complex numbers.exe", "PB_MatrixTranspose.cpp", "PC_CurrencyInformation.", and "PC_CurrencyInformation.". The "All Files (*.*)" filter is selected, and the file "hello.java" is highlighted.

In the center of the IDE, a small command prompt window is open, titled "C:\Windows\system32\cmd.exe". It displays the output of the program:

```
Hello World!
Press any key to continue . . . _
```

The IDE's status bar at the bottom shows "For Help, press F1", "Ln 1", "col 13", "8", "00", "PC", "ANSI", and the system clock "1:06 AM 9/5/2011".

Thank you