Object Oriented Programming

Lecture -1
Introduction

What is Java

- A simple, object-oriented, distributed, interpreted, robust, secure, architecture neutral, portable, high-performance, multithreaded, and dynamic language -- **Sun Microsystems**
- **Object-Oriented**
  - No free functions
  - All code belong to some class
  - Classes are in turn arranged in a hierarchy or package structure

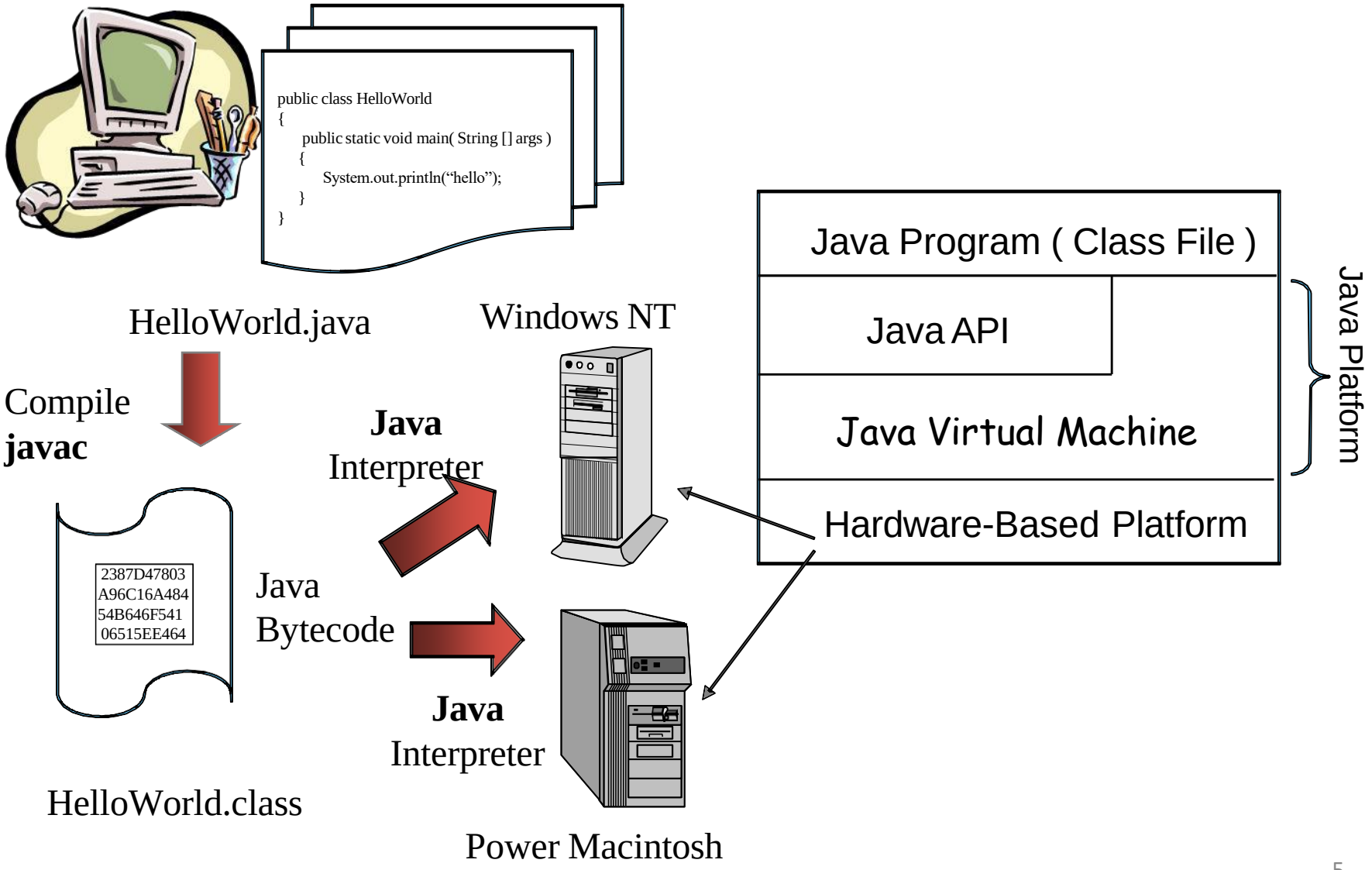
What is Java

- **Interpreted Language**
  - The program are compiled into Java Virtual Machine (JVM) code called bytecode
  - Each bytecode instruction is translated into machine code at the time of execution

What is Java

- **Robust**
  - Java is simple – no pointers/stack concerns
  - Exception handling – try/catch/finally series allows for simplified error recovery
  - Strongly typed language – many errors caught during compilation

Java platform



Java Development Environment

- Edit
  - Create/edit the source code
- Compile
  - Compile the source code
- Execute
  - Execute the compiled

Phase 1: Creating a Program

- Any text editor or Java IDE (Integrated Development Environment) can be used to develop Java programs
- Java source-code file names must end with the ***.java*** extension
- Some popular Java IDEs are
  - IntelliJ
  - NetBeans
  - Eclipse

Phase 2: Compiling a Java Program

- ***javac Welcome.java***
  - Searches the file in the current directory
  - Compiles the source file
  - Transforms the Java source code into bytecodes
  - Places the bytecodes in a file named **Welcome.class**

Bytecodes *

- They are not machine language binary code
- They are independent of any particular microprocessor or hardware platform
- They are platform-independent instructions
- Another entity (interpreter) is required to convert the bytecodes into machine codes that the underlying microprocessor understands
- This is the job of the **JVM** (Java Virtual Machine)

JVM (Java Virtual Machine) *

- It is a part of the JDK and the foundation of the Java platform
- **It can be installed separately or with JDK**
- It is the JVM that makes Java a portable language
- The JVM is invoked by the java command
  - ***java Welcome***

  It searches the class Welcome in the current directory and executes the main method of class Welcome

Phase 3: Execution

- Now the actual execution of the program begins
- Bytecodes are converted to machine language suitable for the underlying OS and hardware

- Java programs go through two compilation phases
  - Source code -> Bytecodes
  - Bytecodes -> Machine language

Editing a Java Program

```
public class Welcome {

    public static void main(String args[]){

        System.out.println("Hello World");
    }


}
```

Examining Welcome.java

- A Java source file can contain multiple classes, but only one class can be a public class
- Typically, Java classes are grouped into packages

- The source file name must match the name of the public class defined in the file with the .java extension

Examining Welcome.java

- In Java, there is no provision to declare a class, and then define the member functions outside the class
- Body of every member function of a class (called method in Java) must be written when the method is declared
- Java methods can be written in any order in the source file