Web Technology 4
Introduction To Java

Chittaranjan Pradhan
School of Computer Engineering,
KIIT University

Motivations for Java

Motivations for Java

- In-built garbage-collection facilities
- Portable facilities
- Multithreading concept
- Designed by James Gosling, Patrick Naughton, Chris Warth, Ed Frank and Mike Sheridan at Sun Microsystems in 1991

Features of Java

- Java is a pure object-oriented language
- Java is completely independent of any particular o.s. platform. As long as the JVM (Java Virtual Machine) is installed, the java program will run identically on all machines
- A java program is compiled & interpreted. The source code is compiled into an intermediate byte code file. This intermediate file is interpreted by a Java interpreter
  - Byte code makes the java program portable across different h/w and o.s. platforms
  - Increased security due to the control of the JVM over the execution of the byte code file

Features of Java...

Features of Java...

- Java provides implicit support to memory management. The automatic technique of freeing unused memory is called garbage collection
- Java has a strong exception-handling mechanism
- Java supports multithreading, i.e. a single process can do multiple tasks apparently simultaneously
- Java supports TCP/IP and UDP protocol families and it can be used for programming in a networked environment

Object-Oriented Programming

- The data is treated as the most important element and it can't flow freely around the system
- **Encapsulation**
  - Class is used as a unit to group related attributes and operations together. The outside world can interact with the data stored in the variables that represent the attributes of the class only through the operations of that class
- **Abstraction & Implementation Hiding**
  - Class is one abstract unit. In order to perform a task that involves an object of that class, a message must be sent to the object asking it to execute the respective operation
  - Implementation hiding means the manner in which data is stored in an object, is hidden from the outside

Object-Oriented Programming...

- **Inheritance, Dynamic Binding and Polymorphism**
  - Inheritance is a property by which one class inherits the features of another class
  - Dynamic binding or runtime binding or late binding is a technique in which the piece of code to be executed is determined only at runtime
  - Polymorphism means the ability to take more than one form
- **Overriding and Overloading**
  - Overriding of a method in a class is the redefinition of a method in the sub-classes of that class
  - Method overloading is a term used when several methods within the same class have the same name but different signatures (signature means the number and type of parameters in the method)
  - *In overriding, the same method name and same signature can be used in different sub-classes of the class and defined differently in each sub-class; whereas in overloading, the same name of the method with different signatures is used multiple times in the same class*

Advantages of Object-Oriented Programming

Advantages of Object-Oriented Programming

- It supports reusability of code
- Frameworks can be created
- It provides a clear modular structure for programs
- Easier for updating of codes in methods

Java Language Features

- **Simple**: Java is designed to be easy for the professional programmer to learn and use
- **Object-oriented**: a clean, usable, pragmatic approach to objects, not restricted by the need for compatibility with other languages
- **Robust**: restricts the programmer to find the mistakes early, performs compile-time (strong typing) and run-time (exception-handling) checks, manages memory automatically
- **Multithreaded**: supports multi-threaded programming for writing program that perform concurrent computations
- **Architecture-neutral**: JVM provides a platform-independent environment for the execution of byte code

Java Language Features...

- **Interpreted and high-performance**: Java programs are compiled into an intermediate representation- byte code:
  - can be later interpreted by any JVM
  - can be also translated into the native machine code for efficiency
- **Distributed**: Java handles TCP/IP protocols, accessing a resource through its URL much like accessing a local file
- **Dynamic**: substantial amounts of run-time type information to verify and resolve access to objects at run-time
- **Secure**: programs are confined to the Java execution environment and can't access other parts of the computer

JDK, JRE & JVM

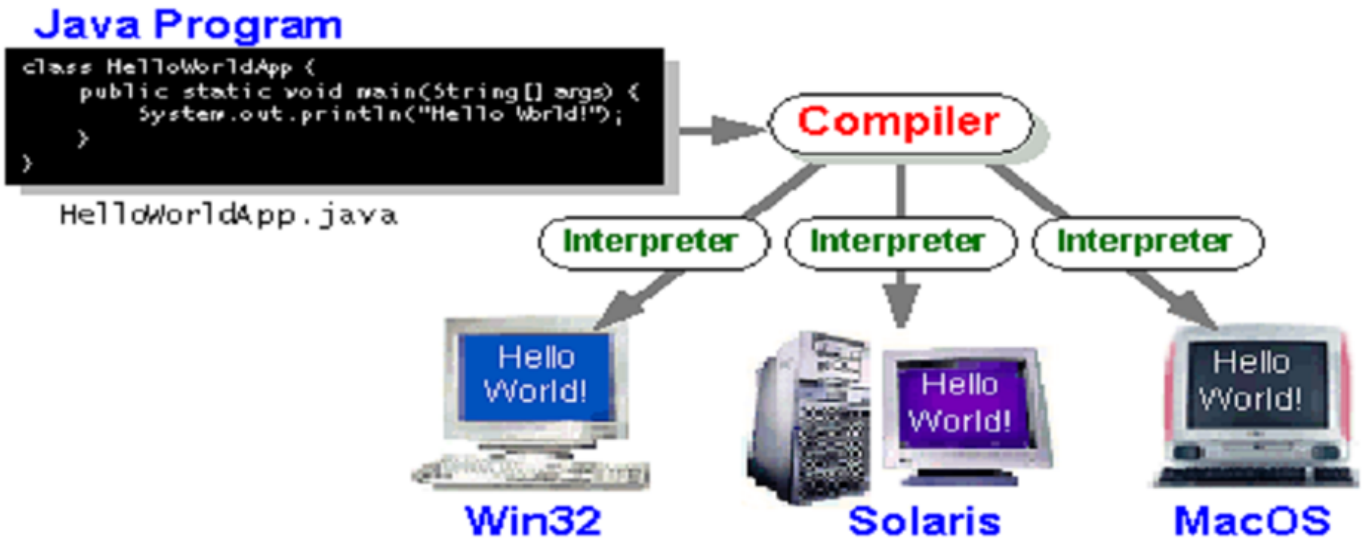
- **JVM** (Java Virtual Machine) is an abstract machine. It is a specification that provides runtime environment in which java bytecode can be executed
  - Loads code
  - Verifies code
  - Executes code
  - Provides runtime environment
- **JRE** (Java Runtime Environment) is used to provide runtime environment. It is the implementation of JVM. It contains set of libraries + other files that JVM uses at runtime
- **JDK** (Java Development Kit) contains JRE + Development tools

Tools in the JDK

Tools in the JDK

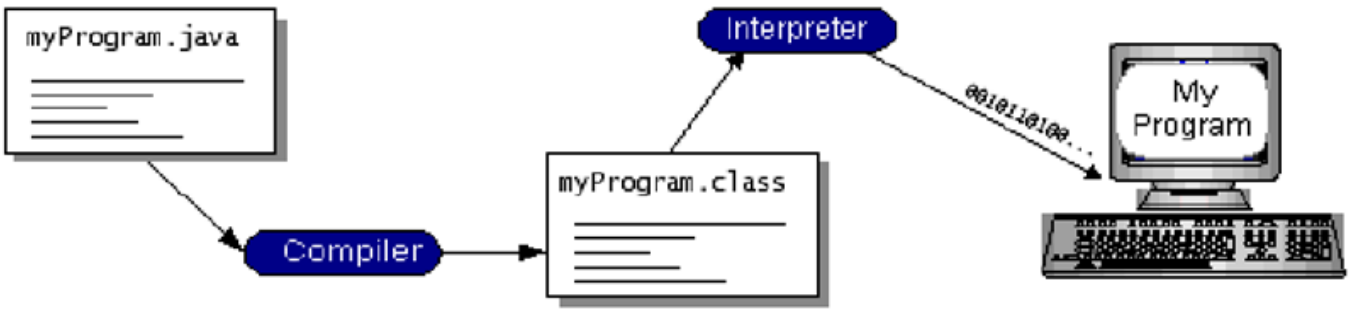
- JDK contains the necessary facilities for the compilation of Java programs into intermediate byte code and their interpretation. It consists of various tools that can be used by the programmer to develop java programs:
  - **javac**: it takes as input a java source code file and produces a class file that contains the byte code
  - **java**: it takes as input, a class file containing the byte code and runs the program by interpreting it
  - **jdb**: it is a debugger that assists the developer in detecting errors
  - **javadoc**: it takes a java source code file as input and produces documentation in html for it
  - **javah**: it takes a java source code file as input and produces header files for use with native methods
  - **javap**: it takes the byte code file as input and produces a file that gives a description of the source code file from which the byte code file was created after compilation. It is a java disassembler
  - **appletviewer**: it is a tool that permits the user to execute java applets without using any java-compatible browser

Java Platform Independence



Java Program Execution

- Java programs are both compiled and interpreted
- Steps:
  - write the Java program
  - compile the program into byte code
  - execute (interpret) the byte code on the computer through the Java Virtual Machine
- *Compilation happens once, while Interpretation occurs each time the program is executed*



Java Technologies

- Java technology is both a programming language and a platform
- Different technologies depending on the target applications:
  - Java Platform, Standard Edition (Java SE)
  - Java Platform, Enterprise Edition (Java EE)
  - Java Platform, Micro Edition (Java ME)
  - Smart Card Applications - JavaCard
- Each edition puts together a large collections of packages offering functionality needed and relevant to a given application
- The Java Virtual Machine remains essentially the same

Simple Java Program

Simple Java Program

A class to display a simple message:

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

- MyProgram is the name of the class
- main is the method with one parameter args and no results
- println is a method whose purpose is to print the argument passed to it to the console and move the cursor to the next line; in the standard System class
- out refers to the standard output stream

Simple Java Program...

- A class is the basic building block of Java programs
- A class encapsulates:
  - data (attributes) and
  - operations on this data (methods)

  and permits to create objects as its instances
- The main method must be present in every Java application
- public static void main(String[] args) where:
  - public means that the method can be called by any object
  - static means that the method is shared by all instances. It is a class method
  - void means that the method does not return any value
- When the interpreter executes an application, it starts by calling its main method which in turn invokes other methods in this or other classes
- The main method accepts a single argument- a string array, which holds all command-line parameters