CHAPTER 3A-INTRODUCTION TO JAVA, JVM, JDK

3a.1 Objective

- Introduction to Java
- History of Java
- Brief to Java Platform, JVM, JDK, JRE
- Installation of JDK and its configurations
- Run a sample program

3a.2 Content

3a.2.1 What is java

Java is a programming language created by James Gosling from Sun Microsystems (Sun) in 1991. The first publicly available version of Java (Java 1.0) was released in 1995.

Sun Microsystems was acquired by the Oracle Corporation in 2010. Now oracle is owning Java.

Java is Simple: There are various features that makes the java as a simple language. because Java is easy to learn and developed by taking the best features from other languages mainly like C and C++.

Java is Platform Independent-Java provides the facility to "Write once -Run any where"(Known as platform independent).

Java is Robust-Java has the strong memory allocation and automatic garbage collection mechanism. It carries out type checking at both compile and runtime making sure that every data structure has been clearly defined and typed. compiler checks the program for any error and interpreter checks any run time error that every data structure is clearly defined and typed. Java manages the memory automatically by using an automatic garbage collector. all the above features makes Java language robust.

3a.2.2 Why Java is important to internet

In a network, two very broad categories of objects are transmitted between the server and your personal computer: passive information and dynamic, active programs. For example, when you read your e-mail, you are viewing passive data. Even when you download a program, the program's code is still only passive data. Even when you download a second type of object can be transmitted to your computer: a dynamic, self-executing program. Such a program is and the server yet initiates active agent on the client computer. For example, the server to display properly the data that the server is sending might provide a program.

As desirable as dynamic, networked programs are, they also present serious problems in the areas of security and portability

Let us begin by writing our first Java program that prints a message "Hello, world!" to the display console, as follows:

- Hello World program in java

Step 1: Write the Source Code: Enter the following source codes using a programming text editor (such as TextPad or NotePad++ for Windows or gedit for UNIX/Linux/Mac) or an Interactive Development Environment (IDE) (such as Eclipse or NetBeans - Read the respective "How-To" article on how to install and get started with these IDEs – to see more about IDE refer the Glossary).

Save the source file as "Hello.java". A Java source file should be saved with a file extension of ".java". The filename shall be the same as the classname - in this case "Hello".

// Write simple Hello Program in Java

```
class HelloWorld  //Start
{
        public static void main(String args[])
                {

                System.out.println("Hello World program in java");
                }// Main methods ends
}// Java class ends
```

Step 2: Compile the Source Code: Compile the source code "Hello.java" into portable bytecode "HelloWorld.class" using JDK compiler "javac". Start a CMD Shell (Windows) or Terminal (UNIX/Linux/Mac) and issue this command:

before run java file you should set the path and classpath

path: to instruct compiler that where your JDK is installed

How to set the path:

set PATH=.;C:\Program Files\Java\jdk1.5.0\bin(for example:

C:\Program Files\Java\jdk1.5.0\bin (Provided your jdk installed in the given path)

See the below screen shots in order to run any java program in command prompt mode

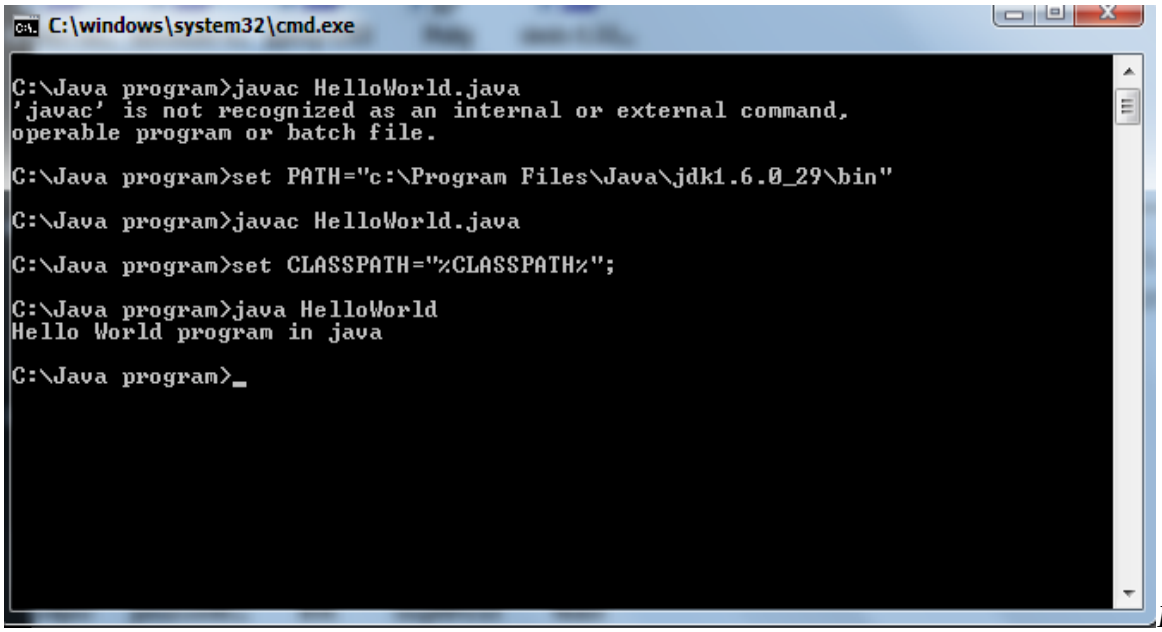

Figure -1 How to run Java Program in Command Prompt Mode

Summary:

Java is platform Independent and Portable.

Java is Robust and secured.

Java is pure object oriented language.

Java does compilation as well Interpretation.

Java latest version of JDK 7(Current version is being used in Industry)

JDK and JRK has to be installed in order to Java program.

3a.2.3 History of Java

Java is released in 1995 as a core component of Sun Micro system's Java platform. In year 1991 they make a platform independent software called it Oak. and in early 1995 Java 1.0 is officially released. This is the first version of Java as JDK1.0. After the first version, many classes, packages are added in Java in future and new version's are released .

3a.2.4 Java version releases

Current version of Java is Java SE7. The Key features in Java SE 7:

- Strings in switch Statement
- Type Inference for Generic Instance Creation
- Multiple Exception Handling
- Support for Dynamic Languages

- Try with Resources
- Java nio Package
- Binary Literals, underscore in literals
- Diamond Syntax
- Automatic Null Handling

Differ Version Available prior to Current Version (Java SE7)

Version	Year of Release	Description
J2SE 1.4	6-Feb-2002	This was the first release of the java platform developed under java community process as JSR 59. The Major Changes are- assert Keyword, Regular Expression, Exception chaining, Image I/O API for reading and writing images, Java Web start included.
J2SE 5.0	30-Sept-2004	New features added are- Generics, Metadata, Autoboxing, Enumeratations, Swing, Var args, collections static imports etc.
Java SE 6	11-Dec-2006	New changes are- Scripting Language support, Improved web service support trough JAX-WS , JDBC 4.0 support, support for pluggable annotations, JVM improvements including Synchronization and compiler performance optimizations, Garbaze collection algorithms

3a.2.5 Java Platform

- ✓ JRE and JDK : See the JDK Section in detail
- ✓ Java Programming Language:
- ✓ Java Virtual Machines – See the JVM Section in detail

- ✓ Base Libraries
- ✓ Integration Libraries
- ✓ User Interface Libraries
- ✓ Deployment
- ✓ Java Deployment
- ✓ Tool Specifications
- ✓ JDK Tools & Utilities
- ✓ Platforms

3a.2.6 JVM- Java Virtual Machine

A Java virtual machine (JVM), an implementation of the Java Virtual Machine Specification, interprets compiled Java binary code (called bytecode) for a computer's processor(or "hardware platform") so that it can perform a Java program's instructions. Java was designed to allow application programs to be built that could be run on any platform without having to be rewritten or recompiled by the programmer for each separate platform. A Java virtual machine makes this possible because it is aware of the specific instruction lengths and other particularities of the platform.

JVM is:

- **A specification** where working of Java Virtual Machine is specified. But implementation provider is independent to choose the algorithm. Its implementation has been provided by Sun and other companies.
- **An implementation** Its implementation is known as JRE (Java Runtime Environment).
- **Runtime Instance** Whenever you write java command on the command prompt to run the java class, and instance of JVM is created.

The JVM performs following operation:

- Loads code
- Verifies code
- Executes code
- Provides runtime environment.

The below diagram will explain in details how JVM converts java file into class and byte code.

JVM Architecture:

3a.2.7 JDK - Java Development Kit

A Java Development Kit (JDK) is a program development environment for writing Java applications. It consists of a run time environment that sits on top of the operating system layer and also the tools and programming that developers need to compile, debug, and run applications written in the Java language.

The JDK (Java Development Kit) is used for developing java applications. The JDK includes JRE, set of API classes, Java compiler, Webstart and additional files needed to write Java applications. The JDK (Java Development Kit) contains software development tools which are used to compile and run the Java program. Both JDK and JRE contains the JVM

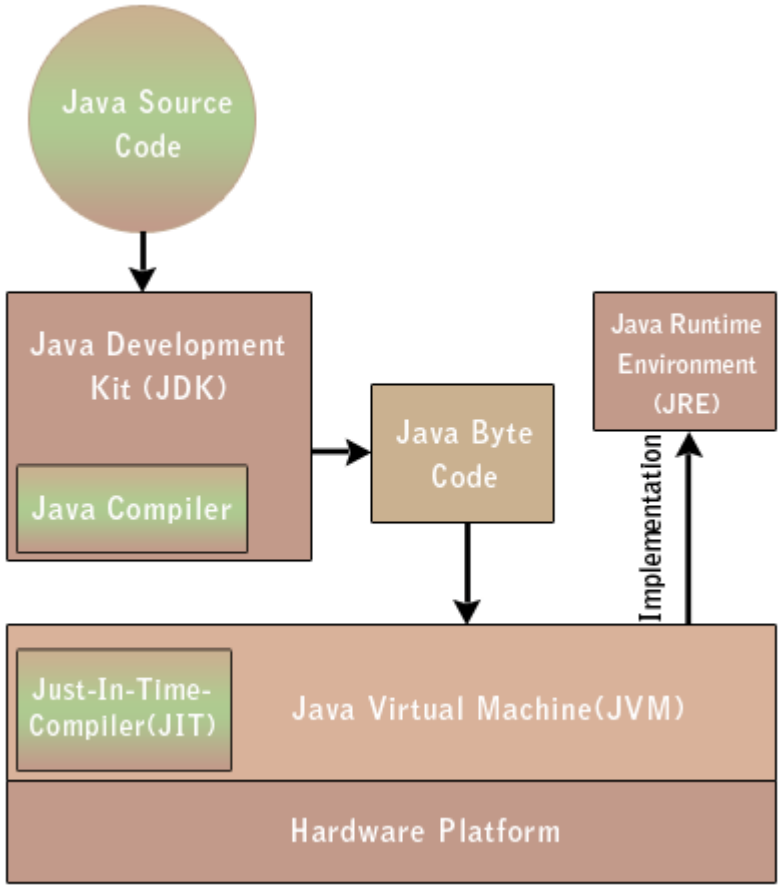

Figure -1 JDK

3a.2.8 JRE- Java Runtime Environment

Java Runtime Environment (JRE), also known as Java Runtime, is part of the JDK. JRE provides the

minimum requirements for executing a Java application; it consists of the Java Virtual Machine (JVM), core classes, and supporting files. Thus it covers most end-users needs and contains everything required to run Java applications on our system.

3a.2.9 Java Environment Setup

Java installation and environment variable setup steps are clearly described in chapter 3b. Follow the instructions given in that chapter and set up the Java environment.

Download JDK 5 or above setup for installation. Link to download latest version of java, jdk 7 is given below.
http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html

Set the environment variables CLASSPATH and JAVA_HOME after JDK installation.

3a.2.10 Know more on Command Promt

1. What is Command Prompt?

Command Prompt is a feature of Windows that provides an entry point for typing MS-DOS (Microsoft Disk Operating System) commands and other computer commands. The most important thing to know is that by typing commands, you can perform tasks on your computer without using the Windows graphical interface.

When you're using Command Prompt, the term command prompt also refers to the right angle bracket (>, also known as the greater than character) that indicates the command line interface can accept commands. Other important information, such as the current working directory (or location) where the command will be run, can be included as part of the command prompt.

For example, if you open the Command Prompt window and see the C:\> command prompt with a blinking cursor to the right of the right angle bracket character (>), you know that the command you enter will be run on the entire C drive of your computer.

2. How do I get to a command prompt?

Open the Command Prompt window by clicking the Start button Picture of the Start button, clicking All Programs, clicking Accessories, and then clicking Command Prompt.

3a.2.11 Execute a Sample Java Program using Command Prompt

Steps to run a java program is described in chapter 3b. We will provide you a sample Java program for execution.

Create a java file “TestClass.java” and put the sample code given below in it. We will study more on how to write a Java class later in this course. This is just to understand how to compile a Java program file and how to execute it using command prompt.

```
class TestClass {
```

```
  public static void main(String args[])
  {
    System.out.println("TestClass is running fine.");
  }
}
```

After execution of TestClass, you should get the message on the command prompt as "TestClass is running fine". Have you got the output on command prompt / console.

3a.3 Reference links

http://www.devmanuals.com/tutorials/java/corejava/JavaHistory.html
http://www.freejavaguide.com/history.html

http://www.devmanuals.com/tutorials/java/corejava/IntroductionJava.html

http://www.w3resource.com/java-tutorial/download-and-Install-JDK-Eclipse-IDE.php

http://docs.oracle.com/javase/tutorial/getStarted/application/index.html

http://www.devmanuals.com/tutorials/java/corejava/JavaEclipseHelloWorld.html
http://www.devmanuals.com/tutorials/java/corejava/JavaVirtualMachine.html
http://viralpatel.net/blogs/java-virtual-machine-an-inside-story/
http://www.javatpoint.com/internal-details-of-jvm#jvminternalarch
http://docs.oracle.com/javase/tutorial/getStarted/intro/definition.html

http://en.wikipedia.org/wiki/Java_Development_Kit
http://www.javatpoint.com/difference-between-jdk-jre-and-jvm

3a.4 Chapter Exit Test