Java Programming Environment

Java development Kit

- JDK or Java Development Kit is software for producing java programs.
- Tools+ classes(are part of JSL java Standard libraries known as API)
- The JDK has as its primary components a collection of programming tools, including:
  - java – the loader for Java applications. This tool is an interpreter and can interpret the class files generated by the javac compiler.
  - javac – the compiler, which converts source code into Java bytecode
  - jar – the archiver, which packages related class libraries into a single JAR file. This tool also helps manage JAR files.
  - javadoc – the documentation generator, which automatically generates documentation from source code comments
  - jdb – the debugger

Con't

- jps – the process status tool, which displays process information for current Java processes
- javap – the class file disassembler
- appletviewer – this tool can be used to run and debug Java applets without a web browser
- javah – the C header and stub generator, used to write native methods
- javaws – the Java Web Start launcher for JNLP applications
- extcheck – a utility which can detect JAR-file conflicts
- apt – the annotation-processing tool [1]
- jhat – (experimental) Java heap analysis tool
- jstack – (experimental) utility which prints Java stack traces of Java threads
- jstat – (experimental) Java Virtual Machine statistics monitoring tool
- jstatd – (experimental) jstat daemon

Con't

- jinfo – (experimental) This utility gets configuration information from a running Java process or crash dump.
- jmap – (experimental) This utility outputs the memory map for Java and can print shared object memory maps or heap memory details of a given process or core dump.
- idlj – the IDL-to-Java compiler. This utility generates Java bindings from a given Java IDL file.
- policytool – the policy creation and management tool, which can determine policy for a Java runtime, specifying which permissions are available for code from various sources
- VisualVM – visual tool integrating several command-line JDK tools and lightweight performance and memory profiling capabilities
- Wsimport – generates portable JAX-WS artifacts for invoking a web service.
- Jrunscript – Java command-line script shell.

Con't

- The JDK also comes with a complete Java Runtime Environment. It consists of a Java Virtual Machine and all of the class libraries.
- Copies of the JDK also include a wide selection of example programs demonstrating the use of almost all portions of the Java API.

Obtaining the Java SDK

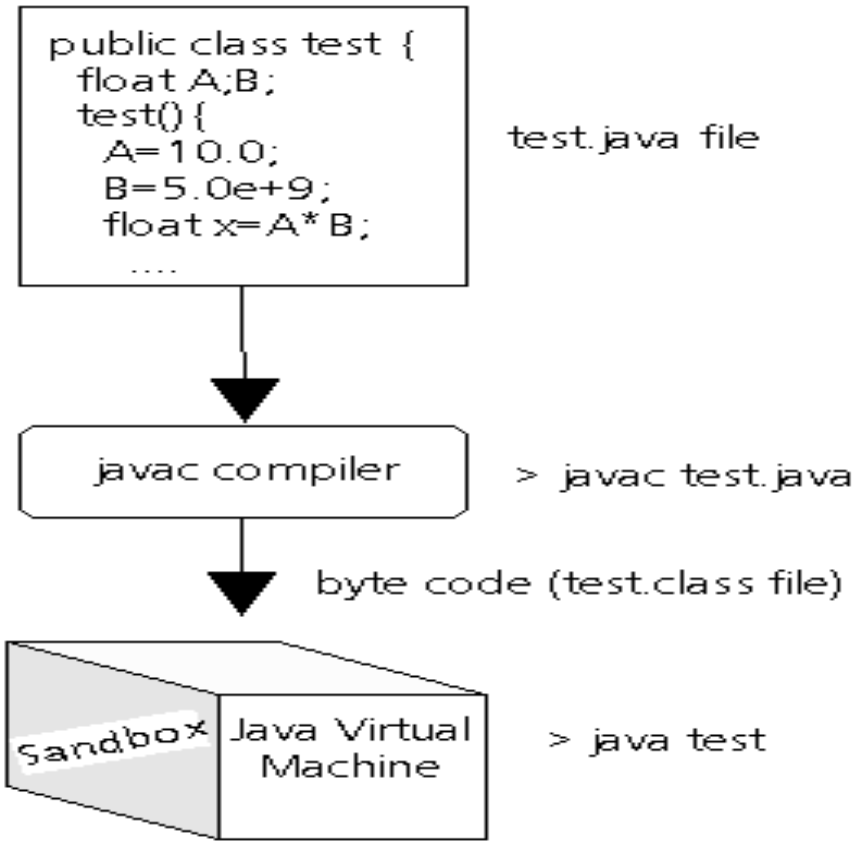
- Download from Sun Web-site:
  - http://java.sun.com/j2se
  - Select "J2SE downloads"
  - Choose your version
  - Select your platform
- Install

- Download java SE Documentation

The Java Procedure

Steps:

- Create a Java source code file
- Compile the source code
- Run the compiled code in a Java Virtual Machine.



Steps for creating and running a Java program.

Programming Tools

Two different *programming environments* for Java:

1) Manual

- Use text editor (see below) create the Java source code files (*.java) and then use the command line tools in the Java *Software Development Kit (SDK)*.
- The SDK is a set of command line tools for developing Java applications:
  - javac - Java Compiler
  - java - Java Interpreter (Java VM)
  - appletviewer - Run applets without a browser
  - javadoc - automated documentation generator
  - jdb - Java debugger
- The SDK is NOT and IDE (Integrated Development Environment)
  - Command line only. No GUI.

Con't

2) Integrated Development Environment (IDE) - graphical user interface programming environments.

- NetBeans - free, open source Java IDE - can be installed along with the SDK.
- Eclipse - free, open source Java IDE
- Borland JBuilder - proprietary but the personal version is free with registration

Java Documentation

Two broad categories:

1) Core Language - defined as the essential set of packages and classes that must be available in the VM regardless of the platform.

2) Optional APIs from Sun - a number of useful *Application Programming Interfaces* are available from Sun such as for audio and video applications, 3D graphics, etc.

- Won't be available for all JVM's on all platforms.

- It is a set of web pages that describe all the elements (packages of classes and the methods).
- for example, what exactly a given method does, what methods are available for a given class, etc.

Java API(Application Programming Interface)

- Java standard library (API) include no. of classes and method grouped several functional packages.
- Most commonly used packages are:

Language (general)	java.lang	Common classes used for all application development
GUI	java.awt java.awt.event javax.swing	Graphical User Interface, Windowing, Event processing
Misc. Utilities and Collections	java.util	Helper classes, collections
Input/Output	java.io	File and Stream I/O
Networking	java.net	Sockets, Datagrams
Applet	java.applet	Applet

Java Program structure

- Documentation section
- Package statement
- Import statements
- Interface statements
- Class definitions
- main method class

Differentiate : C/C++ and JAVA

Difference	C/C++	JAVA
Comment	// /*……*/	// /*……*/
Display function	printf(“ “);	System.out.print(“ ”); System.out.println(“ “);
main()	void main(int argc, char *argv[])	public static void main(String args[])
Header file	#include<stdio.h>	import java.packagename.classname; e.g. import java.lang.String; import java.lang.*;
Compile	Alt+F9	javac filename.java (It generates .class file)
Run	Cltr+F9	If class name and java filename are same java filename If classname and filename are different java classname

Program 1: To display “Hello World”

```
/**
* The HelloWorldApp class implements an application that
* simply displays "Hello World!" to the standard output.
*/
public class HelloWorldApp {
        public static void main(String[] args) {
                System.out.println("Hello World!"); //Display
the string.
        }
}
Compile : javac HelloWorldApp.java
Run       : java HelloWorldApp
Output   : Hello World!
```

public static void main

- Why public?
  - public is access specifier.
  - main() must be declared as public, since it must be called by code outside of its class when the program is started.
- Why static?
  - Static allows main() to be called without having to create particular instance of the class main() is called by java interpreter before any objects are made.
- Why void?
  - Void tells compiler that main does not return a value.

Program 2 : To add two numbers

```
class Addprg{
        public static void main(String args[]){
                int a=10,b=20,c;
                c=a+b;
                System.out.println("Sum is: "+c);
        }
}
```

Compile : javac Addprg.java

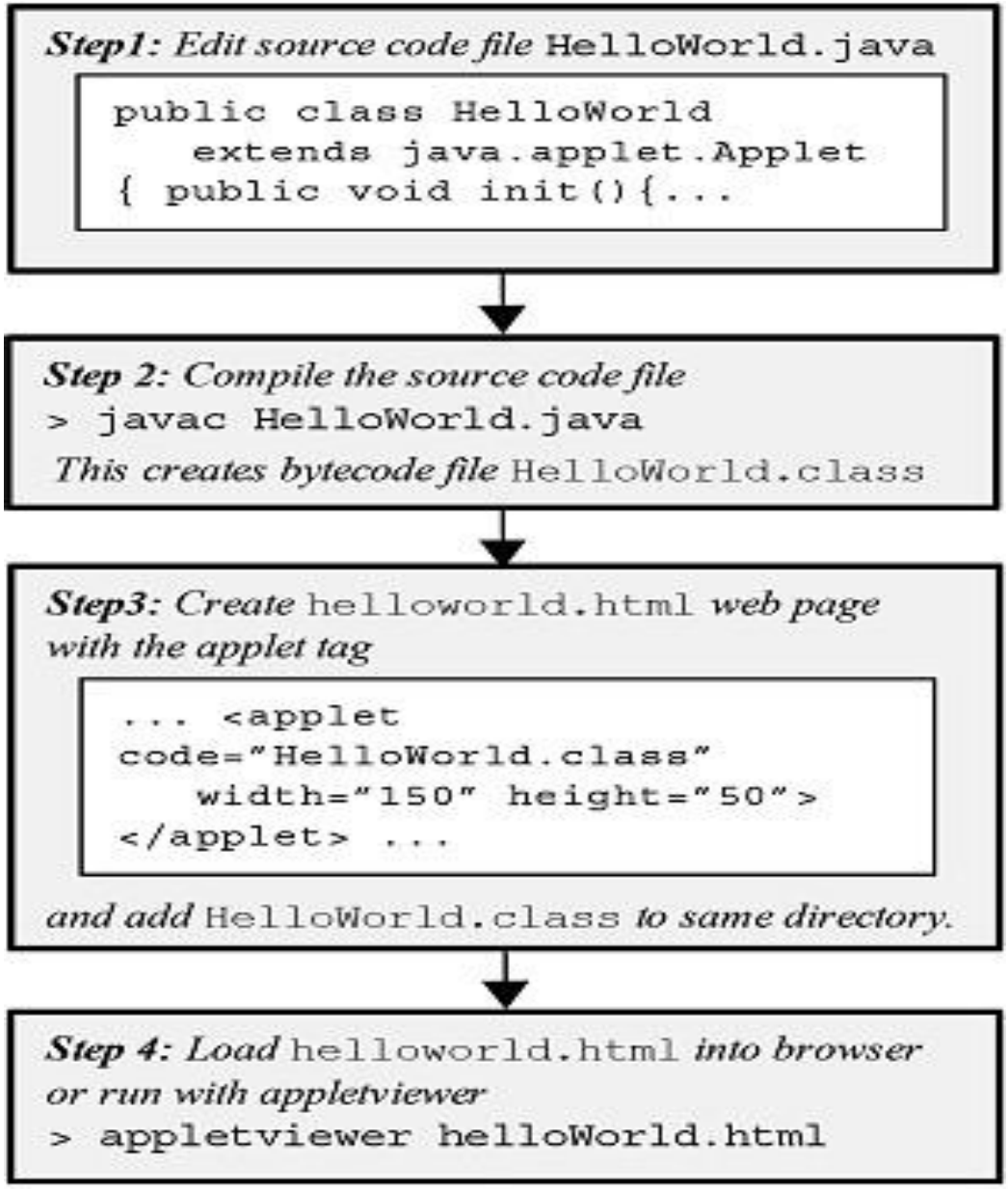
Run : java Addprg

Output : Sum is: 30

Java and www

- Java is strongly associated with internet
  - because the first application program written in Java was Hot java, a web browser to run applets on internet.
  - Applet can be run locally
  - Or can be downloaded from internet and then run on local computer
  - Or can set up websites containing java applets that could be used by other users on internet.
  - Before java, www was limited to display images and texts.
  - Animations, graphics, games were made possible after incorporation of Java into web pages.
  - Hence, web became more interactive and dynamic.
- A Java applet is a small application which is written in Java and delivered to users in the form of bytecode.
- The user launches the Java applet from a web page, and the applet is then executed within a Java Virtual Machine (JVM) in a process separate from the web browser itself.

Creating a Simple Applet



HelloWorld Applet (HelloWorld.java)

```
public class HelloWorld extends java.applet.Applet{
public void paint(java.awt.Graphics g){
        g.drawString("Hello World!",50,25);
        //System.out.println("Hello World!");
}
}
```

HelloWorld.html

```
<HTML>
<Head>
<Title> A Simple Program </Title>
<Body>
<Applet Code="HelloWorld.class" width="150" height="50">
</Applet>
</Body>
</HTML>
```

Steps when java interact with the web

- The user sends a request for HTML document to remote web server which responds the request.
- HTML document contains Applet tag which identifies the applet.
- Applet bytecode is transferred to user's computer. (using javac bytecode has been already created)
- Java-enabled browser on user's computer interprets the bytecodes and provide output.
- User may have further interaction with the applet but with no further downloading from web server. Bytecode contains all necessary information to interpret the applet.