# Object Oriented Programming in Java

**Er.Sital Prasad Mandal**

**BCA- 2nd sem**
**Mechi Campus**
**Bhadrapur, Jhapa, Nepal**
**(Email : info.sitalmandal@gmail.com)**
**https://ctaljava.blogspot.com/**

# Text Book

1. Deitel & Dietel. -Java: How to-program-. 9th Edition. TearsorrEducation. 2011, ISBN: 9780273759168

2. Herbert Schildt. "Java: The CoriviaeReferi4.ic e 61 Seventh Edition. McGraw -Hill 2006, 1SBN; 0072263857

# 1. Introduction to Java

1. Definition
2. History of Java
3. The Internet and Java's Place of IT
4. Applications and Applets
5. Java Virtual Machine
6. Byte Code-not an Executable code
7. Procedure Oriented vs. Object-Oriented Programming
8. Compiling and Running a Simple Program
9. Setting up your Computer for Java Environment
10. Writing a program
11. Compiling
12. Interpreting
13. Running the Program
14. Handling Common Errors.

# Definition

Java is a widely used object-oriented programming language and software platform that runs on billions of devices, including notebook computers, mobile devices, gaming consoles, medical devices and many others. The rules and syntax of Java are based on the C and C++ languages.

One major advantage of developing software with Java is its portability. Once you have written code for a Java program on a notebook computer, it is very easy to move the code to a mobile device. When the language was invented in 1995 by James Gosling of Sun Microsystems (later acquired by Oracle), the primary goal was to be able to "write once, run anywhere.“

It's also important to understand that Java is much different from JavaScript. Javascript does not need to be compiled, while Java code does need to be compiled. Also, Javascript only runs on web browsers while Java can be run anywhere.

# History of Java

- **Java is the object-oriented, platform-independent programming language** used to develop distributed applications that run on the Internet.
- Java was originally called **OAK**.
- Object-Oriented meaning the capability to reuse code.
- Platform-Independent, i.e., we don't have to develop separate applications for different platforms. It is possible to develop a single application that can run on multiple platforms like Windows, UNIX, and Macintosh systems.
- Java designed for hand-held devices and set-top boxes.

**Java Code Analysis**

# History of Java

**Summary in Table format**

| Issue | Statement |
|---|---|
| Designed by | Sun Microsystems |
| Designed In | early 1990s |
| Basic Aim | For Communicating Between Household things |
| Earlier Name of Java | OAK |
| Creator of Java | James Gosling and his team |
| First Public Release | 27 May 1995 |
| Java redesigned for | Internet Development.( Web applications ) |

# History of Java

**Some of the Versions of Java**

| Java Version | Release Date | Year |
|---|---|---|
| JDK 1.0 | January 21 | 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 |

# The Internet and Java's Place of IT

- Java is strongly associated with the internet because of the first application program is written in Java was **Hot Java** (Browser by Sun Micro.).
- Web browsers to run applets on the internet.
- Internet users can use Java to create applet programs & run then locally using a Java-enabled browser such as hot Java.
- Java applets have made the internet a true extension of the storage system of the local computer.

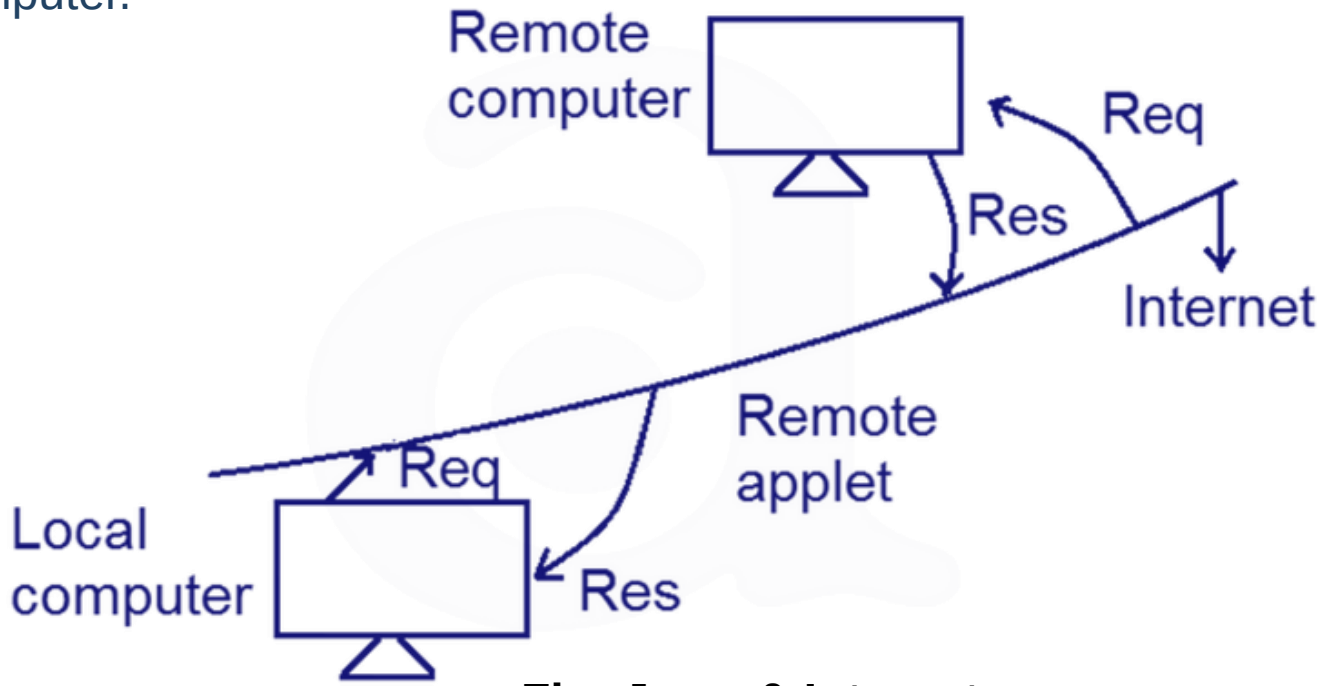



**Fig: Java & Internet**

# The Internet and Java's Place of IT

1. Java communicates with a web page through a special tag called applet .
2. Java user sends a request for an HTML document to the remote computers net browser.
3. The web-browser is a program that accepts a request, processes the request and sends the required documents.
4. The HTML document is returned to that user browser.
5. The document contains the applet tag which identifies the applet. The corresponding applet is transferred to the user computer.
6. The Java enabled browser on the user's computer interprets the byte code and provide output.

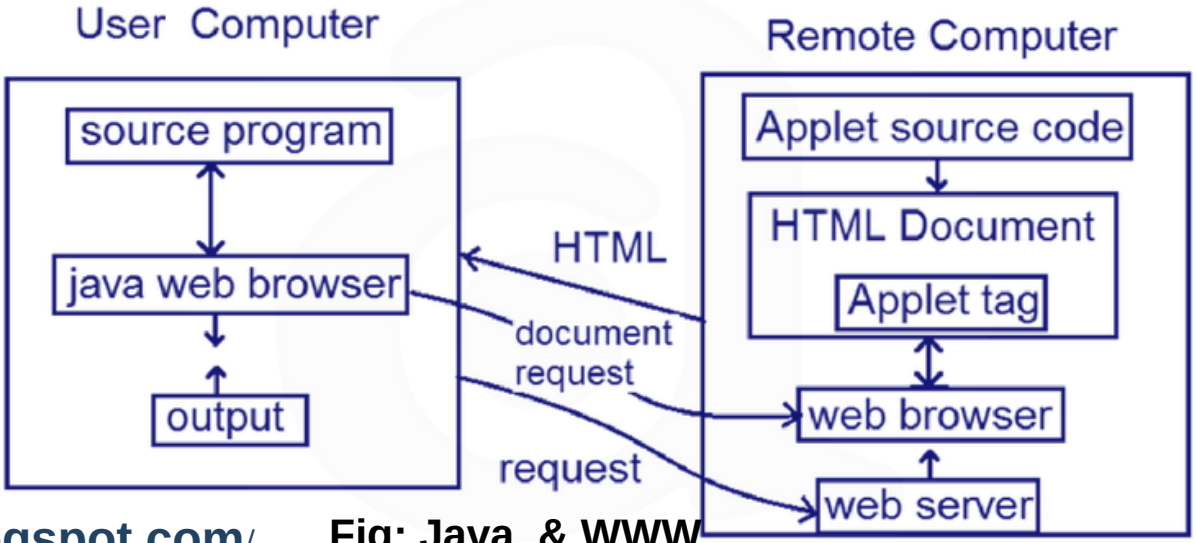



Fig: Java & WWW

# Java Applications and Java Applets

| Feature | Application | Applet |
|---|---|---|
| main() method | Present | Not present |
| Execution | Requires JRE | Requires a browser like Chrome, Firefox, IE, Safari, Opera, etc. |
| Nature | Called as stand-alone application as application can be executed from command prompt | Requires some third party tool help like a browser to execute |
| Restrictions | Can access any data or software available on the system | cannot access any thing on the system except browser's services |
| Security | Does not require any security | Requires highest security for the system as they are untrusted |

# Java Virtual Machine

**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 Oracle 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, an instance of JVM is created.

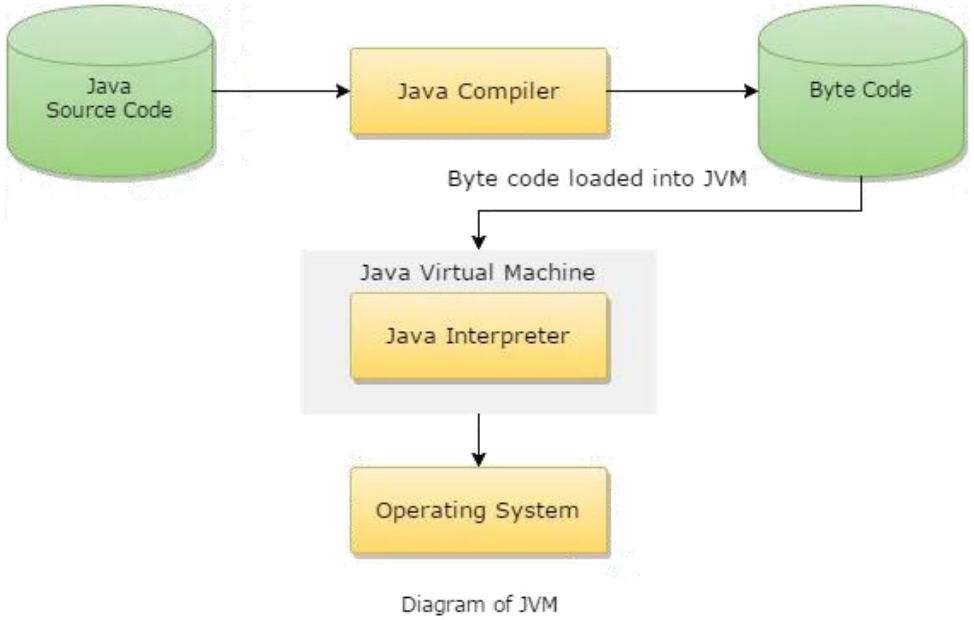
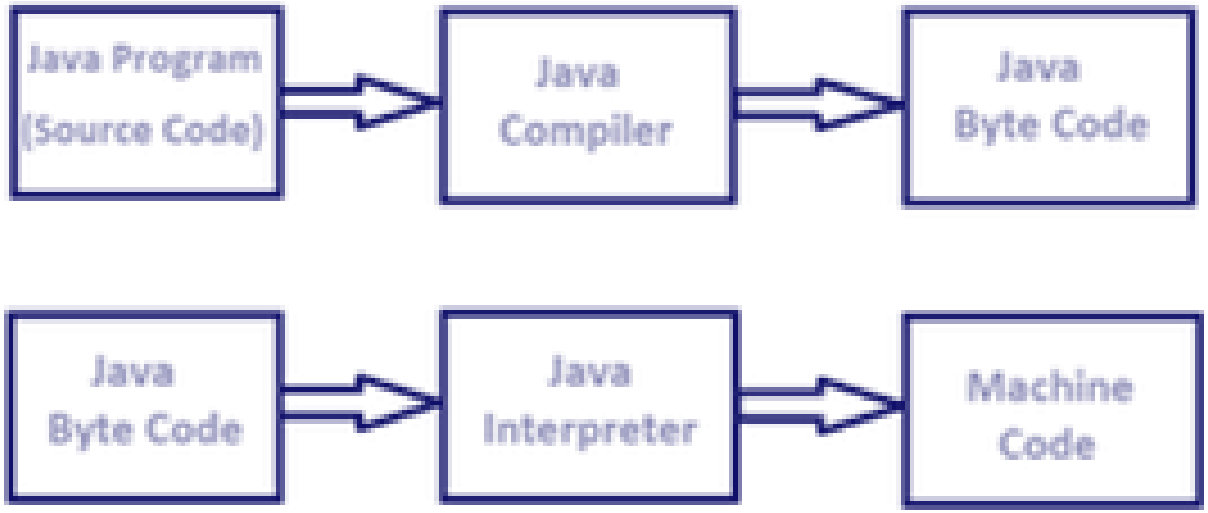



Diagram of JVM

# Byte Code-not an Executable code

Byte code is a non-runnable code after it is translated by an interpreter into machine code then it is understandable by the machine. It is compiled to run on JVM, any system JVM can run it irrespective of their Operating System. That's why Java is platform-independent. Byte code is referred to as a Portable code.



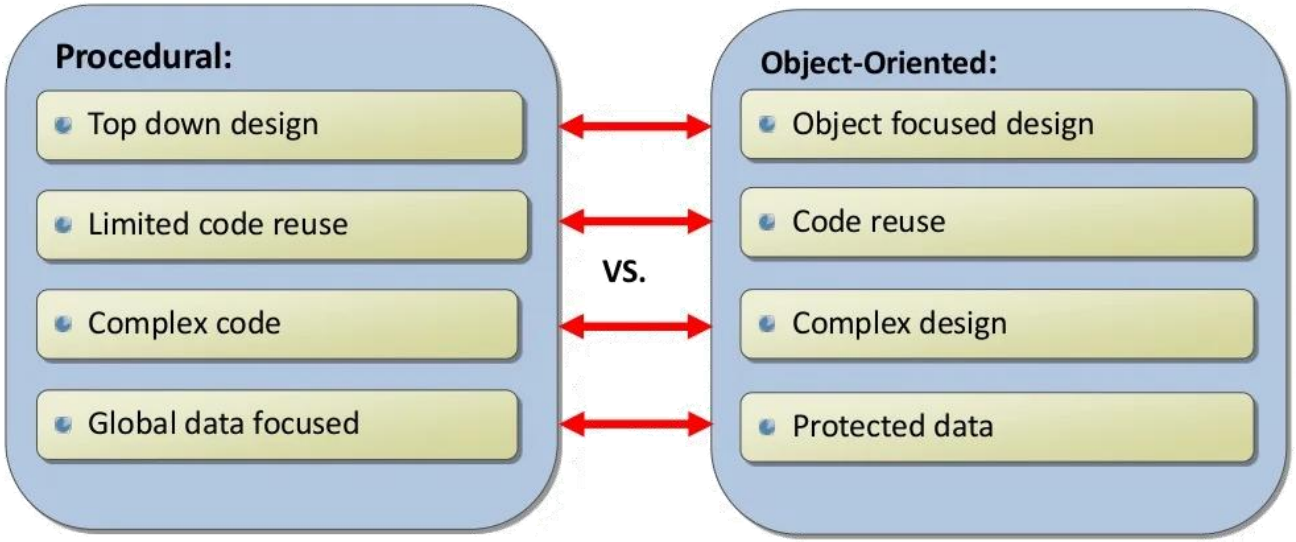
# Assignment

1. Difference between Byte Code and Machine Code.
2. Difference between Source Code and Byte Code.
3. Bytecode vs. Binary Code.

| Bytecode | Binary Code |
|---|---|
| Bytecode is an intermediate level code between the source code and the machine code executed by a virtual machine. | The most simplistic form of data represented by a binary system of digits consisting of a string of consecutive 0's and 1's. |
| Bytecode is considered as the intermediate-level code. | Binary code is considered as the low-level code. |
| Bytecode is compiled to run on a virtual machine (VM) instead of a central processing unit (CPU). | Binary code is directly machine understandable. |
| Bytecode is platform-independent because it is compiled to run on a virtual machine. | Binary code is platform dependent because object code of one platform cannot be run on the same OS. |

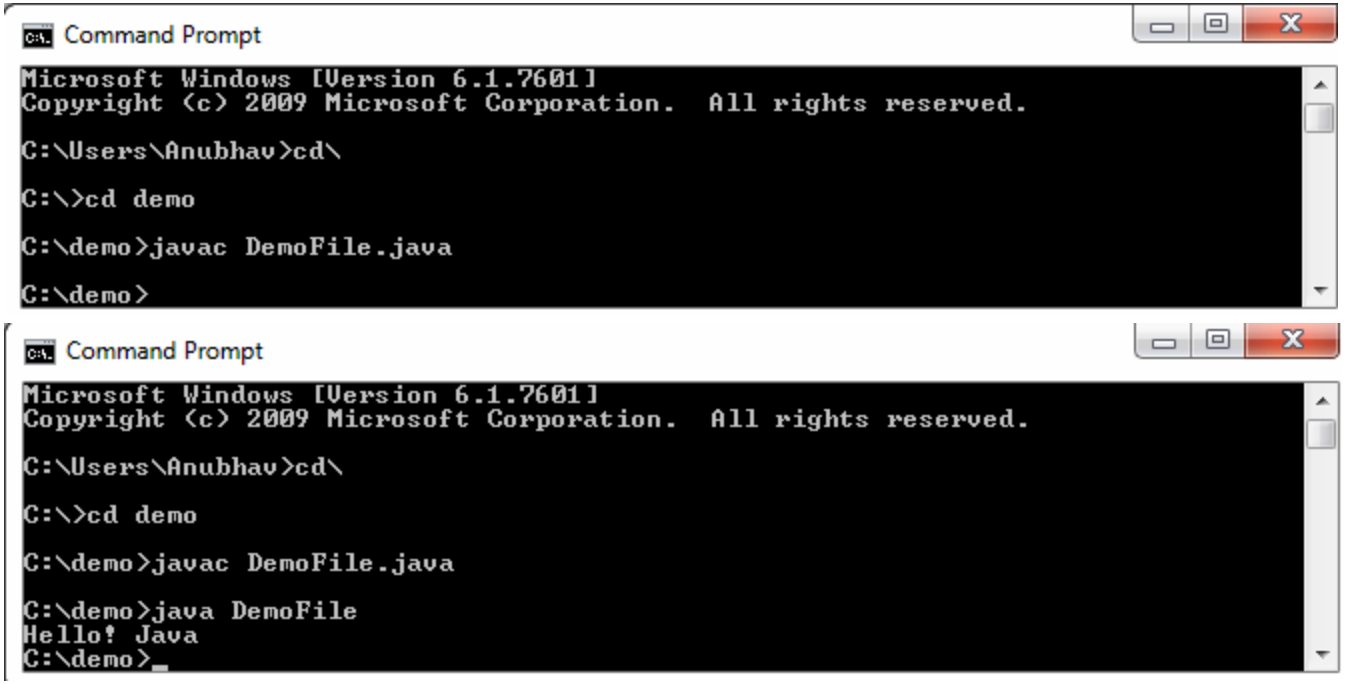
## Procedure Oriented vs. Object-Oriented Programming

| POP | OOP |
|---|---|
| In POP, program is divided into small parts called **functions.** | In OOP, program is divided into parts called **objects.** |
| POP does not have any proper way for hiding data so it is **less** secure. | OOP provides Data Hiding so provides **more** security. |
| Example of POP are : C, VB, FORTRAN, Pascal. | Example of OOP are : C++, JAVA, VB.NET, C#.NET. |



## Compiling and Running a Simple Program

Write a program on the notepad and save it with **.java** (eg. DemoFile.java) extension.

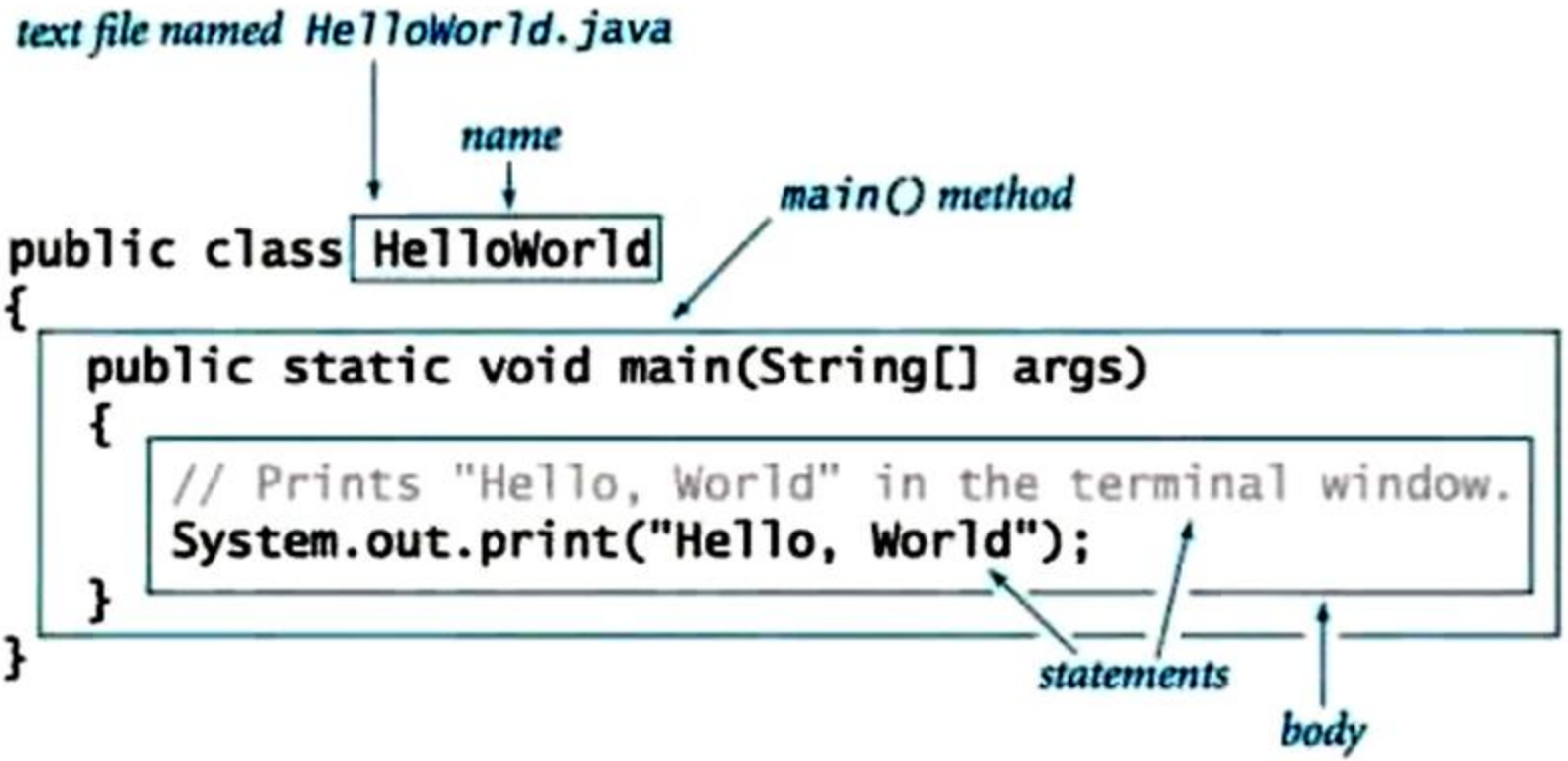



**Set Path in Windows:** set path=C:\Program Files\Java\jdk1.8.0_121\bin
**Set Path in Mac OS X:** export JAVA_HOME=/Library/Java/Home
echo $JAVA_HOME

https://www.oracle.com/java/technologies/downloads/

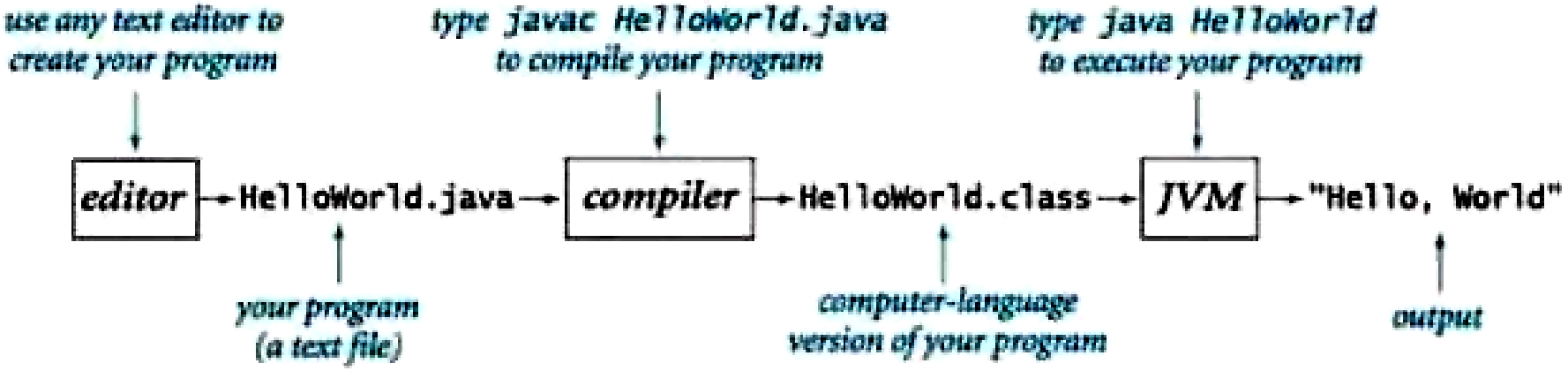
# Simple Java Program

## Hello, World.



# Simple Java Program

## Editing, compiling, and executing.



# Compiled and Interpreter

- **Compiled and Interpreter**: has both Compiled and Interpreter Feature Program of java is First Compiled and Then it is must to Interpret it .
- First of all The Program of java is Compiled then after Compilation it creates Bytes Codes rather than Machine Language.
- Then After Bytes Codes are Converted into the Machine Language is Converted into the Machine Language with the help of the Interpreter.
- So For Executing the java Program First of all it is necessary to Compile it then it must be Interpreter.

***Bytecode is a set of instructions which are designed to be executed by the** Java run time system called the **Java Virtual Machine. Thus the JVM is the** interpreter for the bytecode.*

# Buzzword

**What is an interpreter?**
An interpreter is a software that reads one statement from the source code, translates it to the machine code or virtual machine code, and then executes it right away.

**What is a compiler?**
A compiler is a software that translates a program in high-level language into machine language code.

**APPLET tag**
The HTML tag specifies an applet. It is used for embedding a Java applet within an HTML document. It is not supported in HTML5.

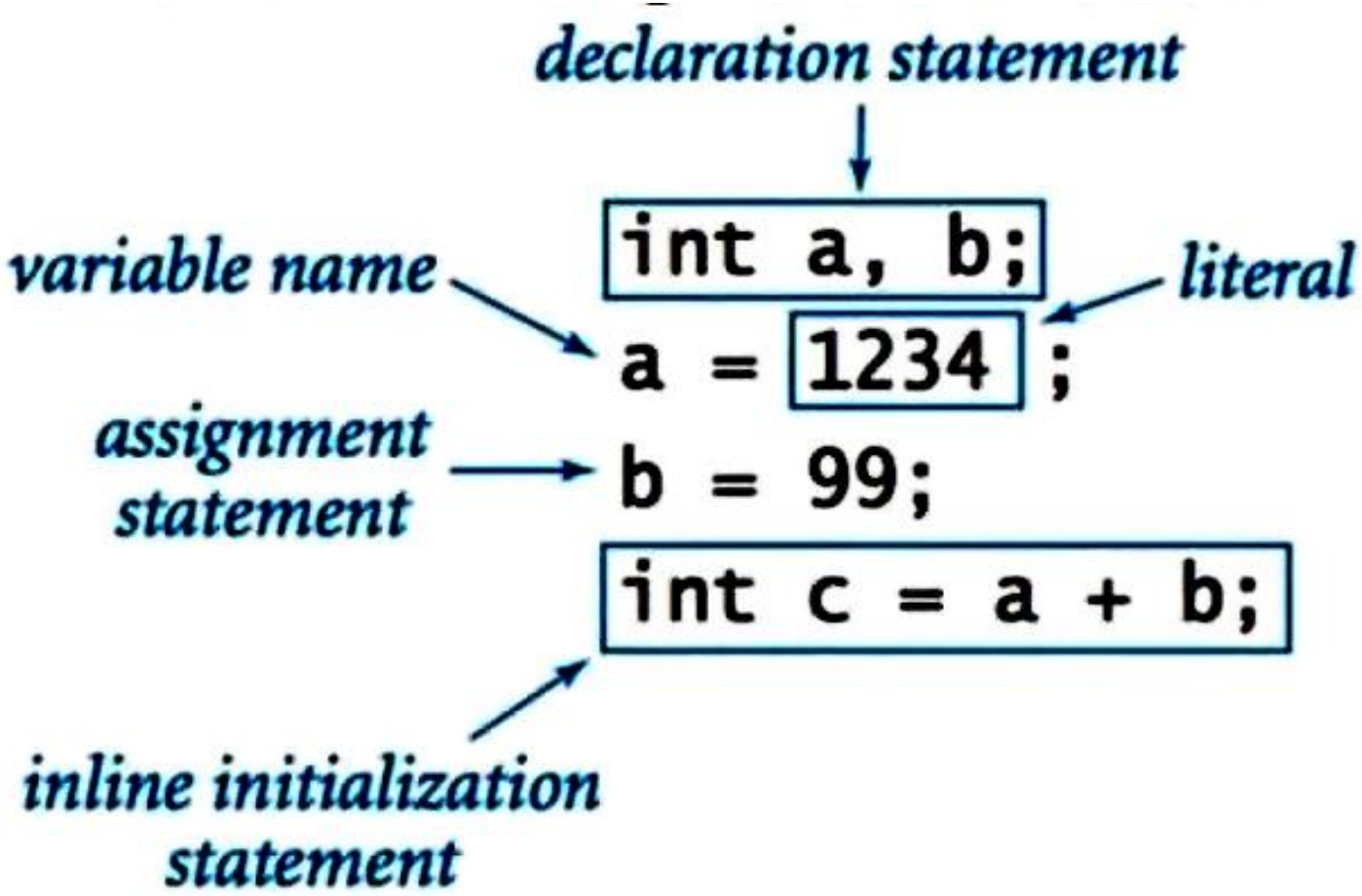
**Java Code**
Java code is used for defining Java APPLETS.

**Byte Code**
Compiled java code that is referred to in the applet tag and transfers to the user computer.

# Built-in Data types

| *type* | *set of values* | *common operators* | *sample literal values* |
|---|---|---|---|
| `int` | integers | `+ - * / %` | `99 12 2147483647` |
| `double` | floating-point numbers | `+ - * /` | `3.14 2.5 6.022e23` |
| `boolean` | boolean values | `&& \|\| !` | `true false` |
| `char` | characters | | `'A' '1' '%' '\n'` |
| `String` | sequences of characters | `+` | `"AB" "Hello" "2.5"` |

# Declaration & Assignment



# Integers

**Integers.**

| | | | | | | |
|---|---|---|---|---|---|---|
| *values* | integers between $-2^{31}$ and $+2^{31}-1$ | | | | | |
| *typical literals* | 1234 99 0 1000000 | | | | | |
| *operations* | *sign* | *add* | *subtract* | *multiply* | *divide* | *remainder* |
| *operators* | + - | + | - | * | / | % |

# Integers

| *expression* | *value* | *comment* |
|---|---|---|
| 99 | 99 | *integer literal* |
| +99 | 99 | *positive sign* |
| -99 | -99 | *negative sign* |
| 5 + 3 | 8 | *addition* |
| 5 - 3 | 2 | *subtraction* |
| 5 * 3 | 15 | *multiplication* |
| 5 / 3 | 1 | *no fractional part* |
| 5 % 3 | 2 | *remainder* |
| 1 / 0 | | *run-time error* |
| 3 * 5 - 2 | 13 | ** has precedence* |
| 3 + 5 / 2 | 5 | */ has precedence* |
| 3 - 5 - 2 | -4 | *left associative* |
| ( 3 - 5 ) - 2 | -4 | *better style* |
| 3 - ( 5 - 2 ) | 0 | *unambiguous* |

# Motivate


We fall. We fail. We break.
But then, we rise. We heal.
We overcome.
@successpictures

## Example

- "reduce, reuse, and recycle" code.

# Reference

https://ecomputernotes.com/java/what-is-java/what-is-java-explain-basic-features-of-java-language

https://self-learning-java-tutorial.blogspot.com/2014/03/features-of-java.html

https://www.atnyla.com/tutorial/java-history/0/5

http://speed.eik.bme.hu/help/html/Web_Programming_Unleashed/ch5.htm

https://pt.slideshare.net/HarisBinZahid/procedural-vs-object-oriented-programming/8