Advance Java Web Development

Java/J2EE

By Bhuwan Prasad Upadhyay

Advance Java Web Development

Week 1

Part-1 : Java Foundation

- Object Oriented Programming
- Abstraction
- Polymorphism
- Inheritance
- Encapsulation
- Introduction of Java
- Features of Java

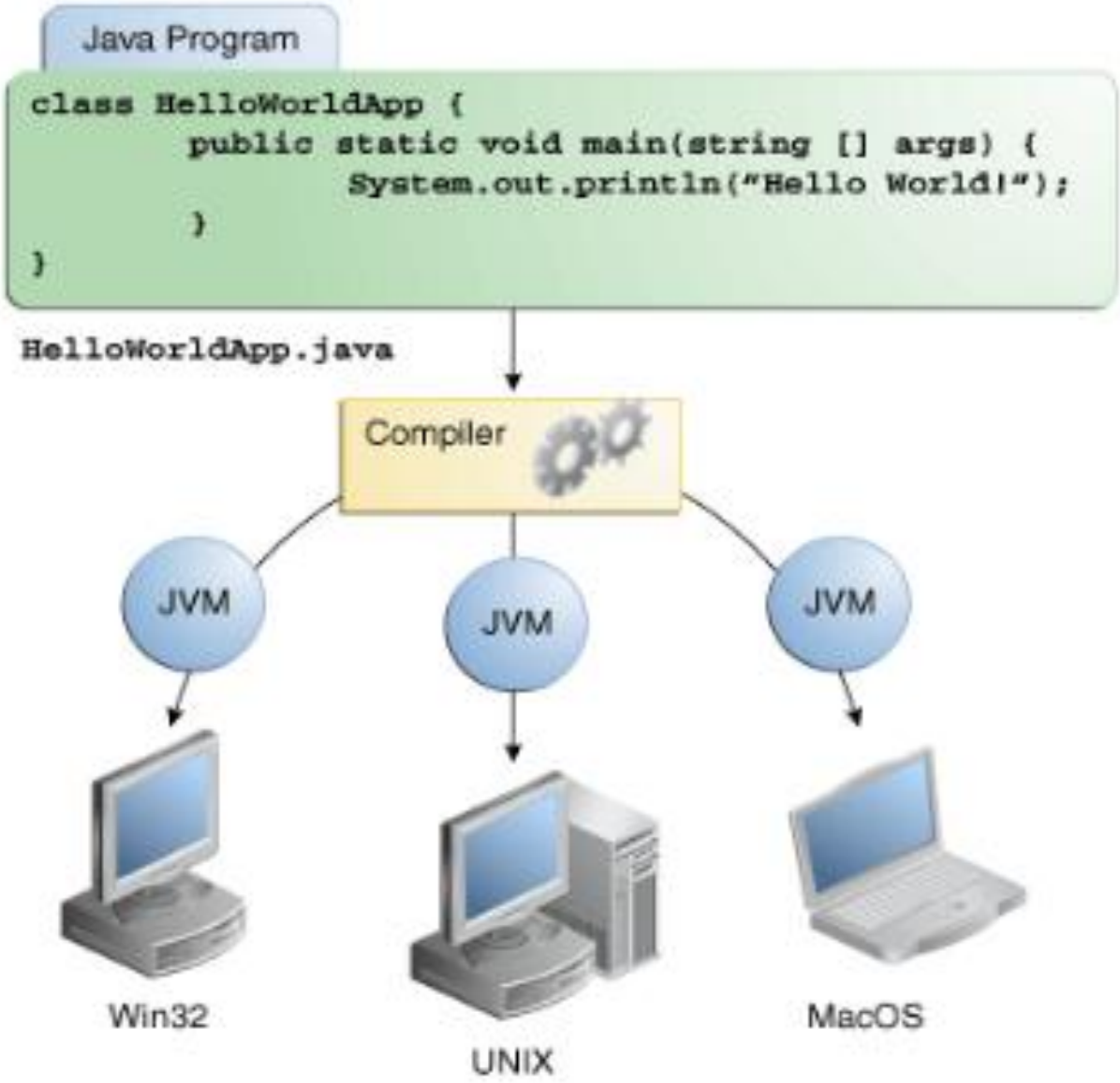
Java Related Terms

- JDK
- JRE
- JVM
- Class Loader
- Compiler
- Interpretator
- Garbage Collector
- Package
- Access Modifiers(VisibilityModes)
- Environment Variables
- Class Path
- Import
- Class
- Object
- Instance
- Byte Code
- JIT- Compilation

Platform Independency

- WORA (Write One Run Anywhere)
- Why Java called Platform Independent ?
- Is Java Application is PI or JVM is PI ?

  PI => Platform Independent



Through the Java VM, the same application is capable of running on multiple platforms.

First Kick on Java

- Install JDK on machine
- Set Java Environment on machine
- Check out java command for environment setup verification
- Write a program to say "I am Java Developer from Now."
- Compile program
- Run Program
- Analyze doing stuff

Part-2 : Java Statements

- Conditional and Looping Statements
- Data Type and Variables in Java
- Primitive/ Derived/User Defined Data Types
- Expression/Operator in Java
- Java String and Array Objects
- Java I/O Streams

Conditional Statements

- if
- If-else
- If-else-if
- switch

Looping Statements

- while
- do-while
- for

Primitive/Derived/User Defined Data Type

- Java programming language is statically-typed, which means that all variables must first be declared before they can be used.
- A primitive type is predefined by the language and is named by a reserved keyword.
- eight primitive data types
- Derived Data Type : int a[] = {1, 2, 3};
- User Defined Data Types:

  Student s = new Student();

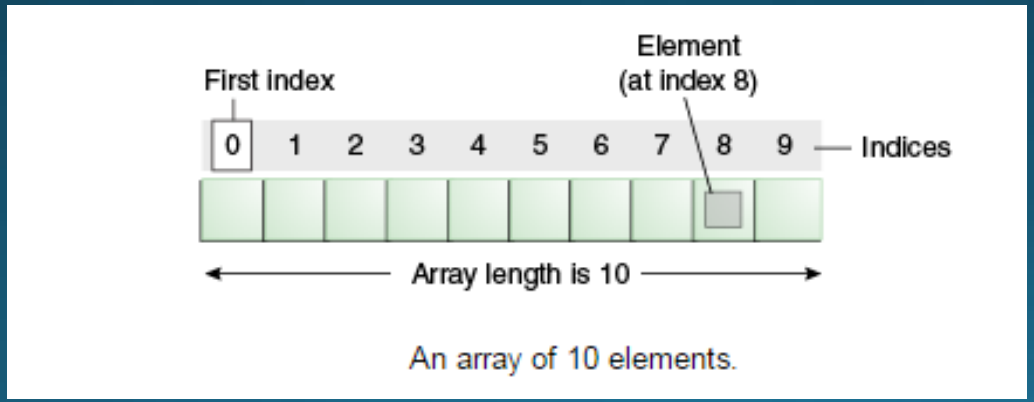
Data Type	Default Value (for fields)
byte	0
short	0
int	0
long	0L
float	0.0f
double	0.0d
char	'\u0000'
String (or any object)	null
boolean	false

Expression/Operator in Java

- Simple Assignment Operator : =
- Arithmetic Operators : +, -, *, /, %
- Unary Operators : +, -, ++, --, !
- Equality and Relational Operators : ==, !=, >, >=, <, <=
- Conditional Operators : &&, ||, ?:
- Type Comparison Operator : instanceof
- Bitwise and Bit Shift Operators : ~, <<, >>, >>>, &, ^, |

Java String and Array Objects

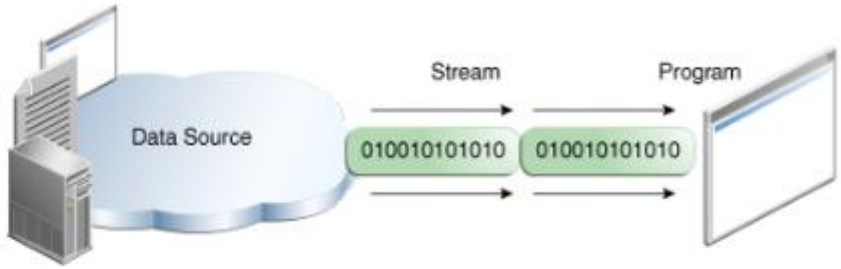
- Java String : Java programming language also provides special support for character strings via the *java.lang.String*
- An *array* is a container object that holds a fixed number of values of a single type. The length of an array is established when the array is created. After creation, its length is fixed.



An array of 10 elements.

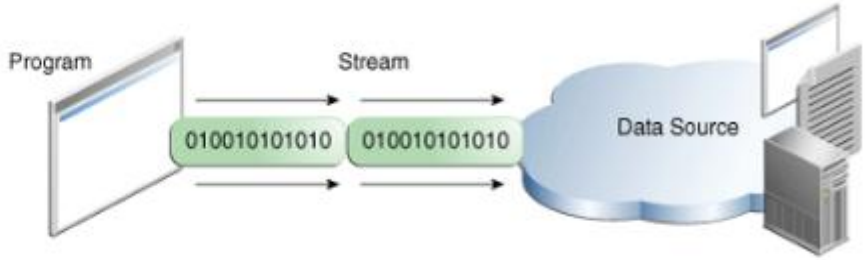
Java I/O Streams

- An *I/O Stream* represents an input source or an output destination. A stream can represent many different kinds of sources and destinations, including disk files, devices, other programs, and memory arrays.



Reading information into a program.

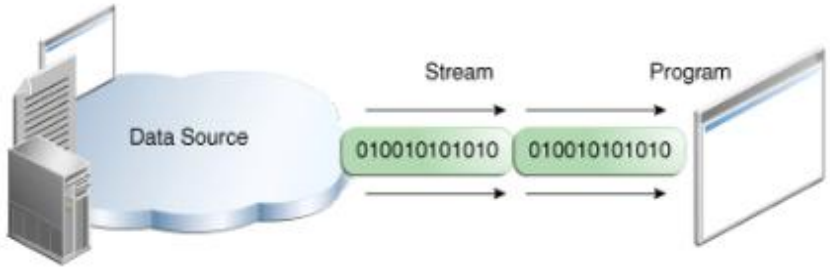
A program uses an *output stream* to write data to a destination, one item at time:



Writing information from a program.

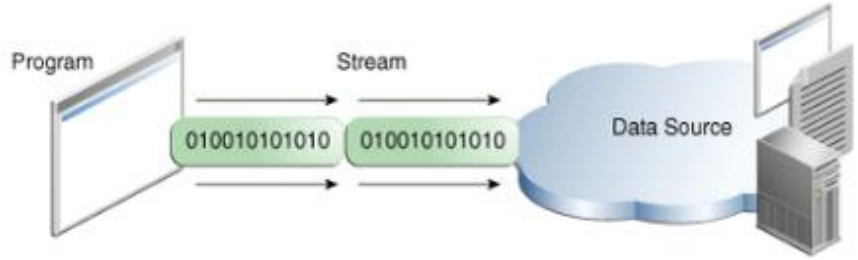
Java I/O Streams

- An *I/O Stream* represents an input source or an output destination. A stream can represent many different kinds of sources and destinations, including disk files, devices, other programs, and memory arrays.
- Byte Stream
- Character Stream
- Buffered Stream



Reading information into a program.

A program uses an *output stream* to write data to a destination, one item at time:



Writing information from a program.

Byte Stream

- perform input and output of 8-bit bytes.
- All byte stream classes are descended from InputStream and OutputStream class

Character Stream

- The Java platform stores character values using Unicode conventions. Character stream I/O automatically translates this internal format to and from the local character set.
- All character stream classes are descended from Reader and Writer.
- Line-Oriented I/O [BufferedReader, PrintWriter]
- Constructor params |-FileReader |-FileWriter

Buffered Reader

- In Java other read or write techniques: request is handled directly by the underlying OS. This can make a program much less efficient, since each such request often triggers disk access, network activity, or some other operation that is relatively expensive.

Java NIO >= jdk1.7

- Path
- Relative vs. Absolute Path
- Symbolic Links

Path Operations

- Creating a Path
- Retrieving Information About a Path

File Operations

- Checking a File or Directory
- Deleting a File or Directory
- Copying a File or Directory
- Moving a File or Directory
- Reading, Writing, and Creating Files
- Finding Files

You Can find me:

Twitter : @DeveloperBhuwan

Portfolio : https://devbhuwan.github.io/

My Blog : https://developerbhuwan.blogspot.com/

End Week-1 : Happy Coding.....