SOFTWARE DESIGN & ANALYSIS (Week-9)

USAMA MUSHARAF

MS-CS (Software Engineering)

LECTURER (Department of Computer Science)

FAST-NUCES PESHAWAR

AGENDA OF WEEK # 9

- Introduction to Java Programming
- Environment Setup
- Syntax and language constructs
- Some Basic Programming
  - Input / Output
  - GUI (Input Output in Dialog Box)
  - Classes and Objects creation
  - Programming with Multiple Classes
  - Functions
  - Constructors

INTRO TO JAVA PROGRAMMING

One of the most widely used computer programming language.

- Java developed by James Gosling at sun micro-systems.
- On January 27, 2010, Sun was acquired by Oracle Corporation for US $7.4 billion, based on an agreement signed on April 20, 2009.
- Pure object oriented language.
- Java is based on the OOP notions of classes and objects.
- Java came on the scene in 1995.

- Before that C and C++ dominated the software development.

Disadvantages of C/C++:

- No garbage collector causes memory leakages.
- Not great support of built in libraries.

JAVA:

- Simple.

  (Very simple syntax similar to c/c++.)
- No operator overloading.
- No multiple Inheritance.
- No pointers.
- Great support of built-in libraries.

- Garbage Collector.
- Programmer Efficiency

  (Building an application in java takes less time than in C/C++)
- Support for Web Application.
- Multi-Threaded.
- Robust/Secure.

- Network Oriented

- Portable/Machine Independent (Write Once Run Anywhere).

Disadvantages:

- C++ is faster than java because java creates intermediate code first and then compile it into target code.

JAVA VIRTUAL MACHINE

When you write a program in C++ it is known as source code.

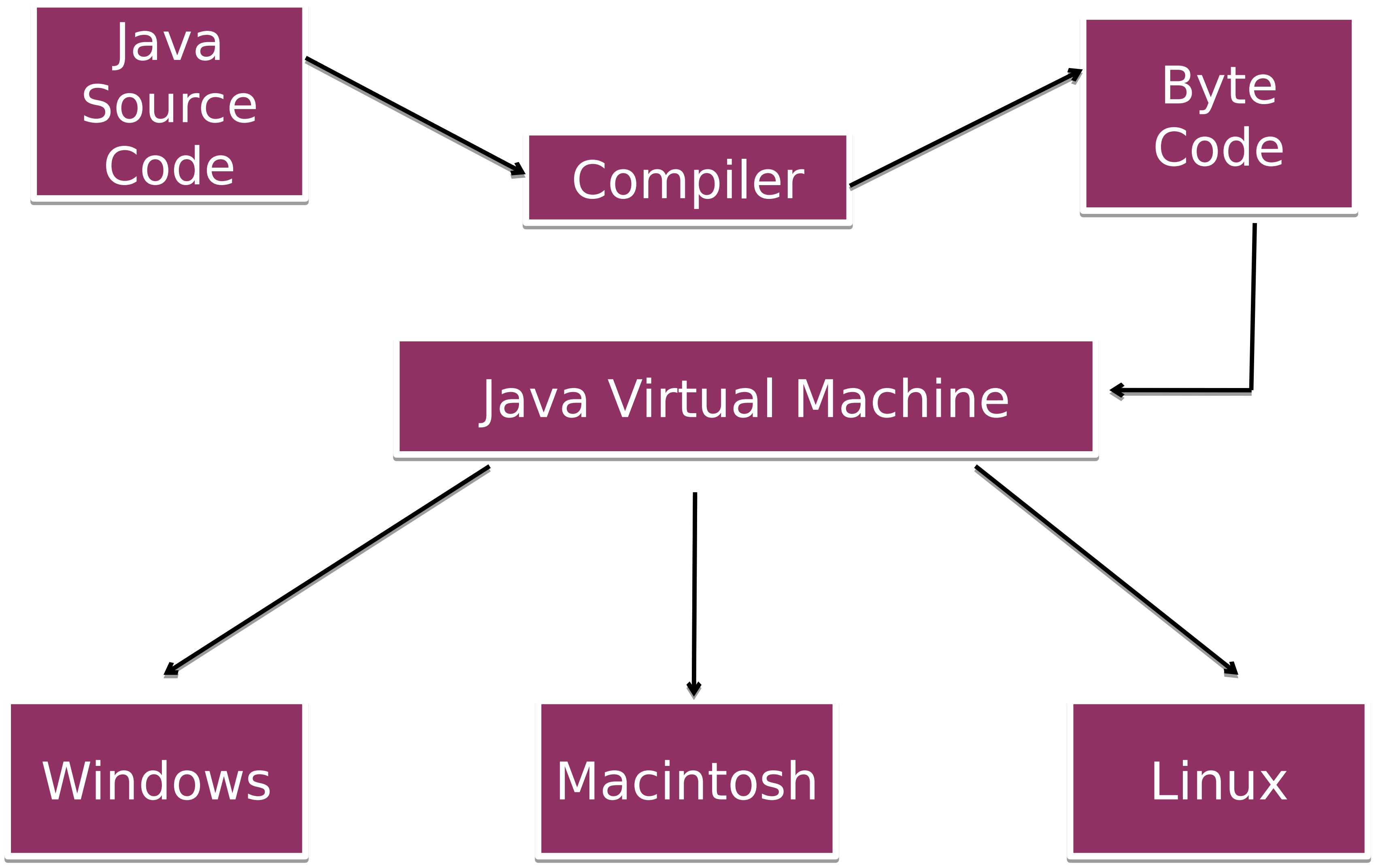
The C++ compiler converts this source code into the machine code of underlying system (e.g. Windows).

If you want to run that code on Linux or on some other operating system you need to recompile it based on the desired compiler.

Due to the difference in compilers, sometimes you need to modify your code.

- Java has the concept of WORA(Write Once Run Anywhere).
- Java compiler does not compile source code for underlying hardware.
- Java compiler compiles a code for system software JVM known as byte code.

- The difference with Java is that it uses byte code - a special type of machine code.
- The JVM executes Java byte codes, so Java byte codes can be thought of as the machine language.
- JVM are available for almost all operating systems.
- Java byte code is executed by using any operating system’s JVM. Thus achieve portability.


Java Source Code
Compiler
Byte Code
Java Virtual Machine
Windows
Macintosh
Linux

Windows Installation

- www.oracle.com ---> downloads ---> Java SE (standard edition)
- Download JDK (Java development kit)   Virtual Machine
- Set path and variables.
- Download Eclipse IDE.

Linux Installation

Download Jdk

```
$ sudo dpkg -i jdk-15_linux-x64_bin.deb
$ sudo update-alternatives --install /usr/bin/java java /usr/lib/jvm/jdk-19/bin/java 1
$ sudo update-alternatives --install /usr/bin/javac javac /usr/lib/jvm/jdk-19/bin/javac 1

$ sudo update-alternatives --config java
$ sudo gedit /etc/profile
Type: JAVA_HOME=/usr/lib/jvm/jdk-19

$ source /etc/profile
```

Download Eclipse IDE.

First Simple Java Program:

```
public class Simple {

public static void main( String args [ ] )
    {
System.out.println("Hello World");
    }
}
```

OUTPUT:

```
Hello World
```

public static void main (string args [])

main() is the function from which your program starts.

“void” indicates that main() function does not return anything.

Way of specifying input (often called command-line arguments) at startup of application.

Why public?

It is kept public so that it is accessible from outside.

Remember private methods are only accessible from within the class.

Why static?

Every Java program starts when the JRE (Java Run Time Environment) calls the main method of that program.

If main is not static then the JRE have to create an object of the class in which main method is present and call the main method on that object (In OOP based languages method are called using the name of object if they are not static).

It is made static so that the JRE can call it without creating an object. Also to ensure that there is only one copy of the main method per class.

JAVA BASIC SYNTAX

When we consider a Java program, it can be defined as a collection of objects that communicate via invoking each other's methods.

- **Object** − Objects have states and behaviors. Example: A dog has states - color, name, breed as well as behavior such as wagging their tail, barking, eating. An object is an instance of a class.

- **Class** − A class can be defined as a template/blueprint that describes the behavior/state that the object of its type supports.

- **Methods** − A method is basically a behavior. A class can contain many methods. It is in methods where the logics are written, data is manipulated and all the actions are executed.

- **Instance Variables** − Each object has its unique set of instance variables. An object's state is created by the values assigned to these instance variables.

- **Case Sensitivity** − Java is case sensitive, which means identifier **Hello** and **hello** would have different meaning in Java.

- **Class Names** − For all class names the first letter should be in Upper Case. If several words are used to form a name of the class, each inner word's first letter should be in Upper Case.
- **Example:** *class MyFirstJavaClass*

- **Method Names** − All method names should start with a Lower Case letter. If several words are used to form the name of the method, then each inner word's first letter should be in Upper Case.
- **Example:** *public void myMethodName()*

- **Program File Name** − Name of the program file should exactly match the class name.

- When saving the file, you should save it using the class name (Remember Java is case sensitive) and append '.java' to the end of the name (if the file name and the class name do not match, your program will not compile).

- **Example:** Assume 'MyFirstJavaProgram' is the class name. Then the file should be saved as '*MyFirstJavaProgram.java*'

- **public static void main(String args[])** − Java program processing starts from the main() method which is a mandatory part of every Java program.

JAVA IDENTIFIERS

Java Identifiers:

- All Java components require names. Names used for classes, variables, and methods are called **identifiers**.
- In Java, there are several points to remember about identifiers. They are as follows.
- All identifiers should begin with a letter (A to Z or a to z), currency character ($) or an underscore (_).
- After the first character, identifiers can have any combination of characters.
- A key word cannot be used as an identifier.
- Most importantly, identifiers are case sensitive.
- Examples of legal identifiers: age, $salary, _value, __1_value.
- Examples of illegal identifiers: 123abc, -salary.

JAVA BASIC DATA TYPES

Primitive data types are not objects in java.

Types of Numeric Variables

- Integer Variables
- Floating-Point Variables

Integer Data types:

The four integer types in java are

- Byte
- Short
- int
- long

Note: When variable of type ***long*** is declared, then the value assigned to the variable will be appended by L.

e.g.   long value=243567L;

Floating-Point Variables:

float

double

Note: The value of type ***float*** will be appended by f.

Character Variable:

Variable of type character stores a single character. 2 bytes space is reserved in memory because all characters in java are stored in Unicode.

e.g   char ch = ‘#’;

Boolean Variable:

Variables that can have one of two values ‘true’ or ‘false’.

e.g boolean flag = false;

Note: Boolean variables cannot be cast to any other type of variable and vice versa.

Casting in Data Types:

- We can cast any of the basic type to any other.
- Casting from the larger integer type to a smaller causes loose information.
- Casting from float to integer loose some information.
- Casting from double to float also loose information.

Byte -- > short -- > int -- > long -- > float -- > double

Left to right : implicitly

Right to left : explicitly

If x = 3 and y=2, then

z= 1 + x/y; Result z=2

z= 1 + (float)x/y Result z =2.5

Assignment Operator (=)

Assignment operator is used to assign literal or value of variable or expression to a variable that comes on its left side.

e.g. c = a*b;

Multiple assignment in java is valid i.e.

a = b = c =100;

Arithmetic Operators

+ , - , *, /, %

e.g. if a = 15 and b = 6 then

a % b = 3

Increment and Decrement Operator

++ and −−

e.g. If x =10, y=5 ,then

Z = x++ +y = 15 and after evaluation, x =11

but in

Z= ++x + y = 16 and after evaluation, x=11

Both can be used as prefix and postfix.

Relational Operators

>, < , <= , >= , == , !=

Relational Expression

If relational operator is applied on operands, then relational expression is formed. The value of relational expression is either true or false.

Logical Operators:

&&	Logical AND
||	Logical OR
!	Logical NOT

Compound Assignment operator:

+=	c+=1	=	c=c+1
*=	c*=1	=	c=c*1
-=	……	=	……..
/=	……	=	……..
%=	……	=	……..

Decision Control Structures:

if – Statement

General syntax is

```
 If (expression)
{
 Statement(s);
}
```

For example

```
If (numbr%2 == 0)
System.out.println(“The number is even”);
```

if-else Statement

General Syntax

```
if (expression)
 {
        statement(s);
}
else
{
       statement(s);
}
```

If-else-if statement

```
if (expression)
 { statements; }

else if (expression)
{ statements; }

else if (expression)
{ statements; }

!
!

else
{ Statements; }
```

Switch Statement

```
switch(variable/expression)
{
   case value1:
   Statements;
   break;
   case value2:
   Statements;
   break;
   case value3:
   Statements;
   break;
   !
   !
   !
   default:
   Statements;
   break;
}
```

STRINGS

A string is commonly considered to be a sequence of characters stored in memory and accessible as a unit.

String Concatenation:

“+” operator is used to concatenate strings.

System.out.pritln(“Hello” + “World”) will print Hello World.

int i = 4;   int j = 5;

System.out.println (“Hello” + i) will print Hello 4 on screen.

However

System.out.println(i+j);

will print 9 on the console because both i and j are of type int.

Lets Code

HAVE A GOOD DAY!