Course Name - Object Oriented Programming using Java

Lecture 6 – Basic concepts of java programming - Advantages of java, Byte-code & JVM, Data types, Different types of Variables..

Presented By
Dr. Sudipta Sahana
Asso Prof.
Dept. of CSE
UEM - Kolkata

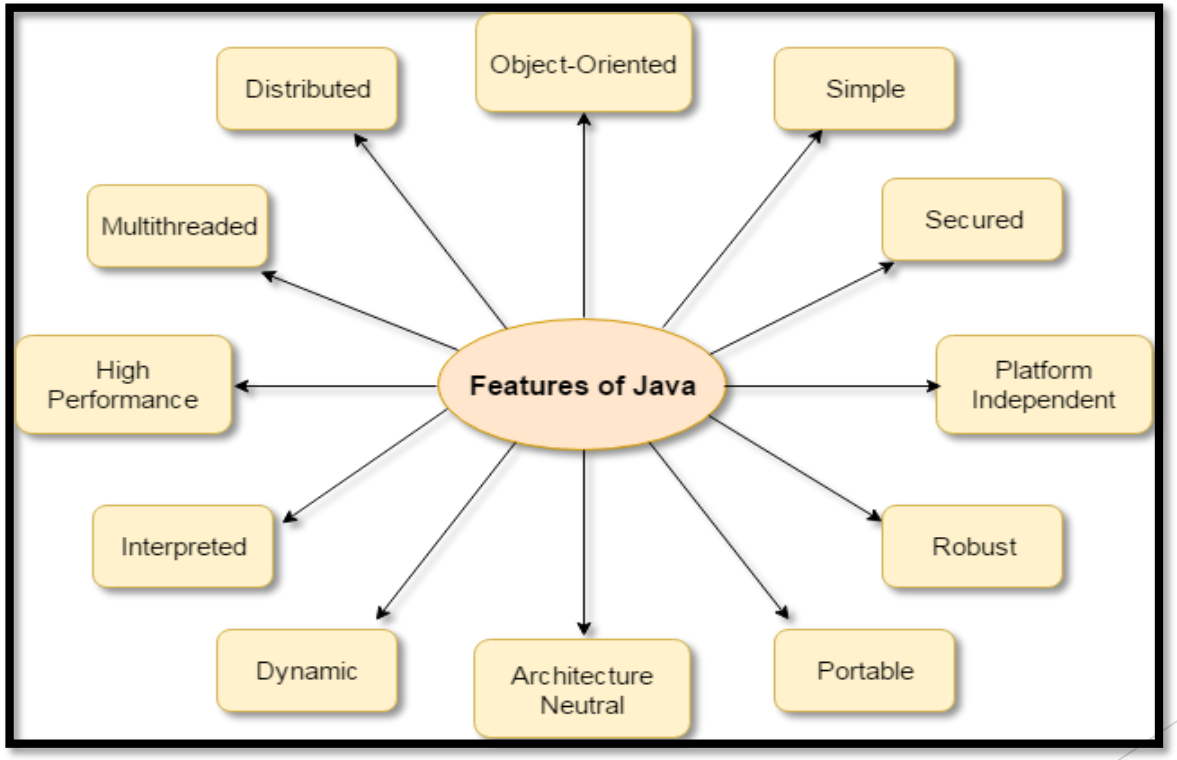
Topic of Interest

- Basic concepts of java
- Byte-Code
- JVM
- Internal Architecture of JVM
- Data types in JAVA
- Different types of Variables

Basic concepts of java programming

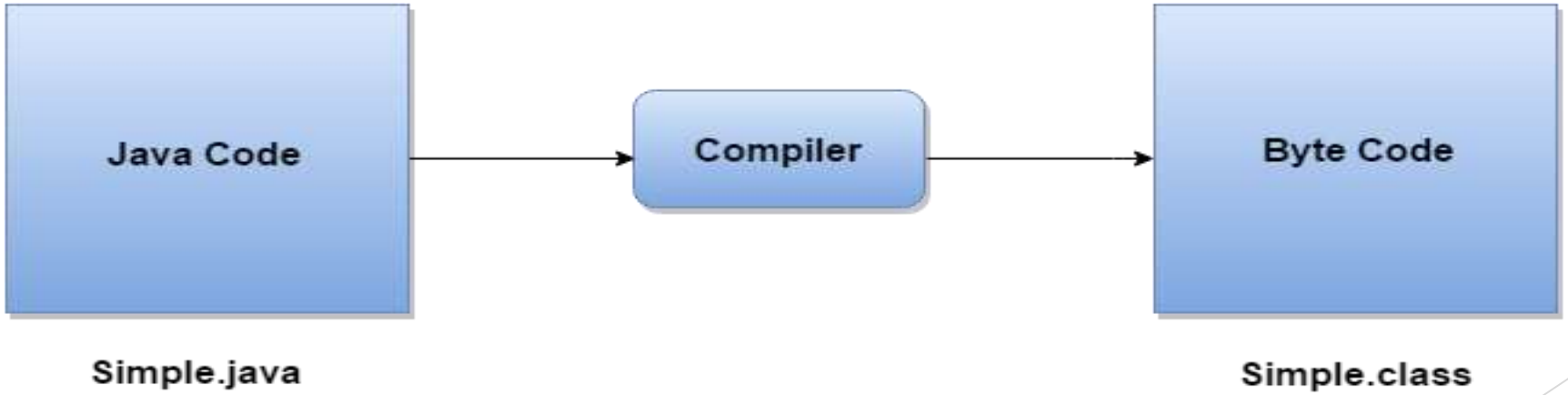
Features of JAVA

- Simple
- Object-Oriented
- Portable
- Platform independent
- Secured
- Robust
- Architecture neutral
- Dynamic
- Interpreted
- High Performance
- Multithreaded
- Distributed



Byte-Code

Byte-code in Java is the reason java is platform-independent, as soon as a JAVA program is compiled byte-code is generated. To be more precise a Java byte-code is the machine code in the form of a .class file.
A byte-code in Java is the instruction set for Java Virtual Machine and acts similar to an assembler.



JVM

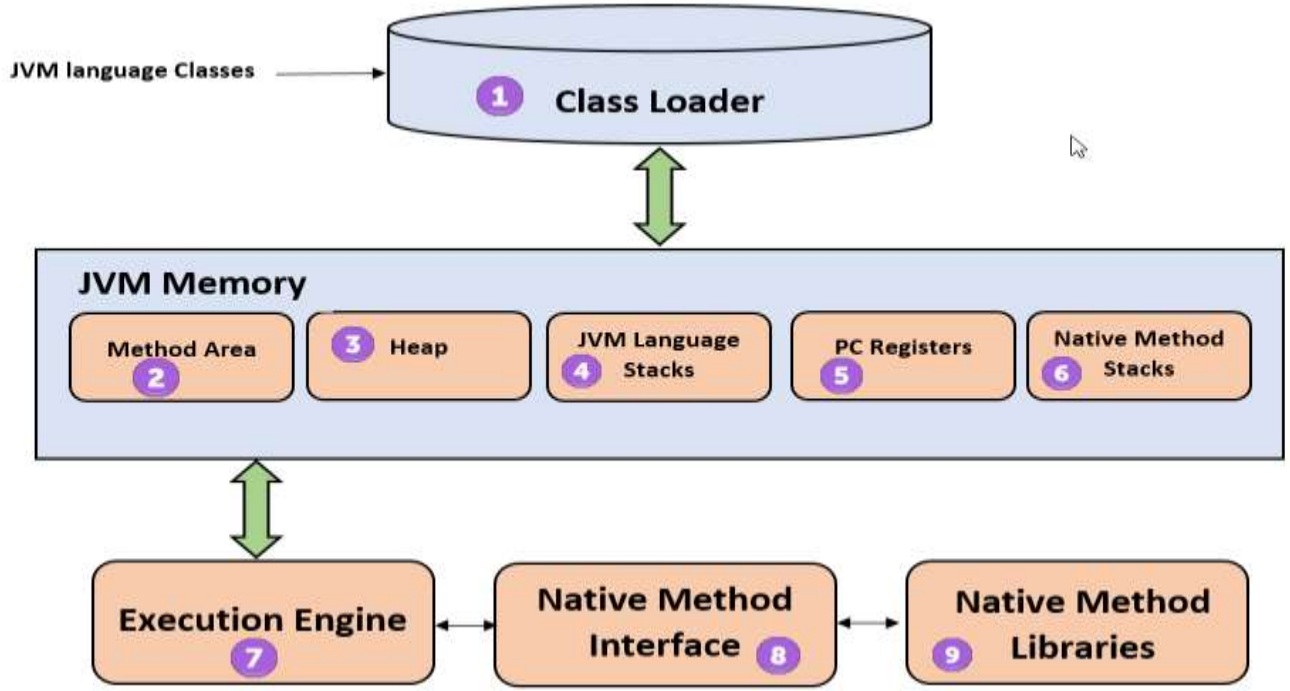
JVM (Java Virtual Machine) is an abstract machine. It is a specification that provides runtime environment in which java byte-code can be executed.

The JVM performs following main tasks:

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

Internal Architecture of JVM:

Let's understand the internal architecture of JVM. It contains class loader, memory area, execution engine etc.



1) Class Loader

The class loader is a subsystem used for loading class files. It performs three major functions viz. Loading, Linking, and Initialization.

2) Method Area

JVM Method Area stores class structures like metadata, the constant runtime pool, and the code for methods.

Internal Architecture of JVM:

3) Heap

All the Objects, their related instance variables, and arrays are stored in the heap. This memory is common and shared across multiple threads.

4) JVM language Stacks

Java language Stacks store local variables, and it's partial results. Each thread has its own JVM stack, created simultaneously as the thread is created. A new frame is created whenever a method is invoked, and it is deleted when method invocation process is complete.

5) PC Registers

PC register store the address of the Java virtual machine instruction which is currently executing. In Java, each thread has its separate PC register.

6) Native Method Stacks

Native method stacks hold the instruction of native code depends on the native library. It is written in another language instead of Java.

7) Execution Engine

It is a type of software used to test hardware, software, or complete systems. The test execution engine never carries any information about the tested product.

8) Native Method interface

The Native Method Interface is a programming framework. It allows Java code which is running in a JVM to call by libraries and native applications.

9) Native Method Libraries

Native Libraries is a collection of the Native Libraries(C, C++) which are needed by the Execution Engine.

Data types in JAVA

There are two data types available in Java −

- Primitive Data Types
- Reference/Object Data Types

There are eight primitive data types in Java:

Data Type	Size	Description
byte	1 byte	Stores whole numbers from -128 to 127
short	2 bytes	Stores whole numbers from -32,768 to 32,767
int	4 bytes	Stores whole numbers from -2,147,483,648 to 2,147,483,647
long	8 bytes	Stores whole numbers from -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
float	4 bytes	Stores fractional numbers. Sufficient for storing 6 to 7 decimal digits
double	8 bytes	Stores fractional numbers. Sufficient for storing 15 decimal digits
boolean	1 bit	Stores true or false values
char	2 bytes	Stores a single character/letter or ASCII values

Reference Data types:

- Reference variables are created using defined constructors of the classes. They are used to access objects. These variables are declared to be of a specific type that cannot be changed. For example, Employee, Puppy, etc.
- Class objects and various type of array variables come under reference datatype.
- Default value of any reference variable is null.
- A reference variable can be used to refer any object of the declared type or any compatible type.
- Example: Animal animal = new Animal("giraffe");

Different types of Variables:

A variable provides us with named storage that our programs can manipulate. Each variable in Java has a specific type, which determines the size and layout of the variable's memory; the range of values that can be stored within that memory; and the set of operations that can be applied to the variable.

- Local variables
- Instance variables
- Class/Static variables

Local Variables:

- Local variables are declared in methods, constructors, or blocks.
- Local variables are created when the method, constructor or block is entered and the variable will be destroyed once it exits the method, constructor, or block.
- Access modifiers cannot be used for local variables.
- Local variables are visible only within the declared method, constructor, or block.
- Local variables are implemented at stack level internally.
- There is no default value for local variables, so local variables should be declared and an initial value should be assigned before the first use.

Instance variables

- Instance variables are non-static variables and are declared in a class outside any method, constructor or block. As instance variables are declared in a class, these variables are created when an object of the class is created and destroyed when the object is destroyed.
- Unlike local variables, we may use access specifiers for instance variables. If we do not specify any access specifier then the default access specifier will be used.
- Initialization of Instance Variable is not Mandatory. Its default value is 0
- Instance Variable can be accessed only by creating objects.

Static Variables:

- Class variables also known as static variables are declared with the static keyword in a class, but outside a method, constructor or a block.
- There would only be one copy of each class variable per class, regardless of how many objects are created from it.
- Static variables are rarely used other than being declared as constants. Constants are variables that are declared as public/private, final, and static. Constant variables never change from their initial value.

UEM
UNIVERSITY OF ENGINEERING & MANAGEMENT
Good Education, Good Jobs

Thank You