SNS COLLEGE OF ENGINEERING

Kurumbapalayam (Po), Coimbatore – 641 107

An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

COURSE NAME : 23CSB101- OBJECT ORIENTED PROGRAMMING

I YEAR /II SEMESTER

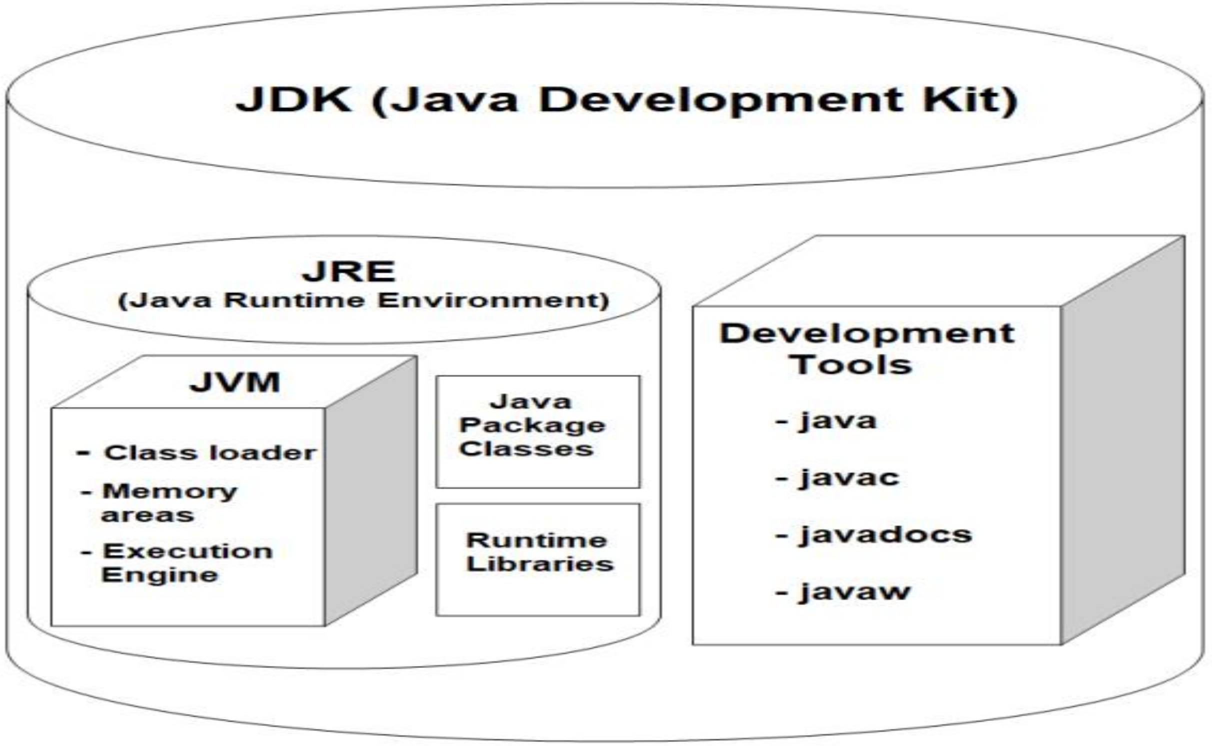
Unit I – INTRODUCTION TO OOP AND JAVA

Topic : CLASSES

Java Architecture

- **Java Development Kit (JDK)** – Includes the compiler, libraries, and tools to develop Java applications.
- **Java Runtime Environment (JRE)** – Provides libraries and JVM to run Java applications.
- **Java Virtual Machine (JVM)** – Converts Java bytecode into machine code for execution.

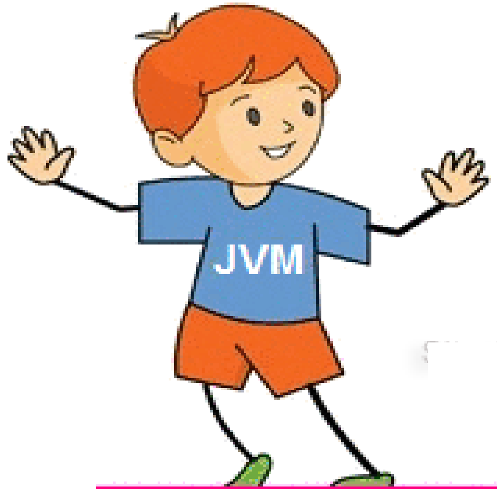
Java Architecture



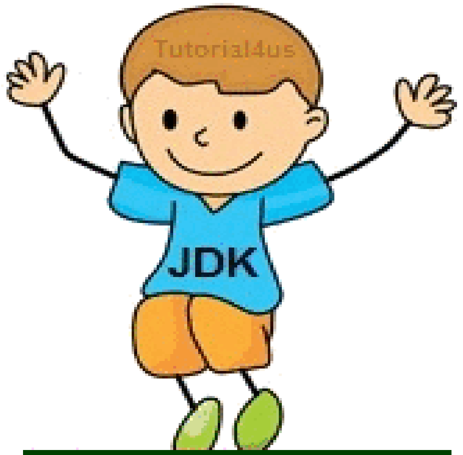
Architecture of JVM, JRE & JDK

JDK = JRE (JVM + API) + DEVELOPMENT TOOLS

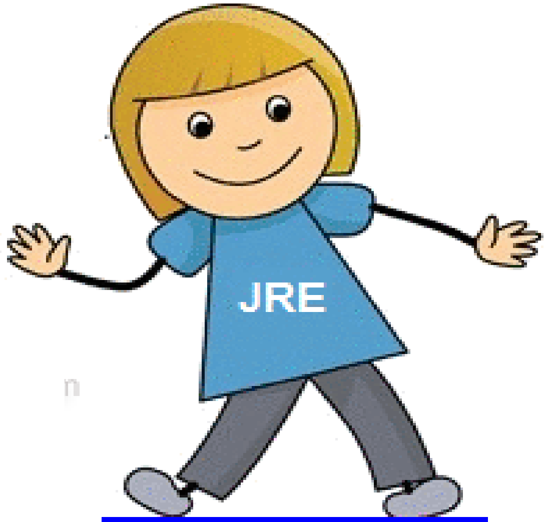
Architecture



I provides runtime environment to execute bytecode.



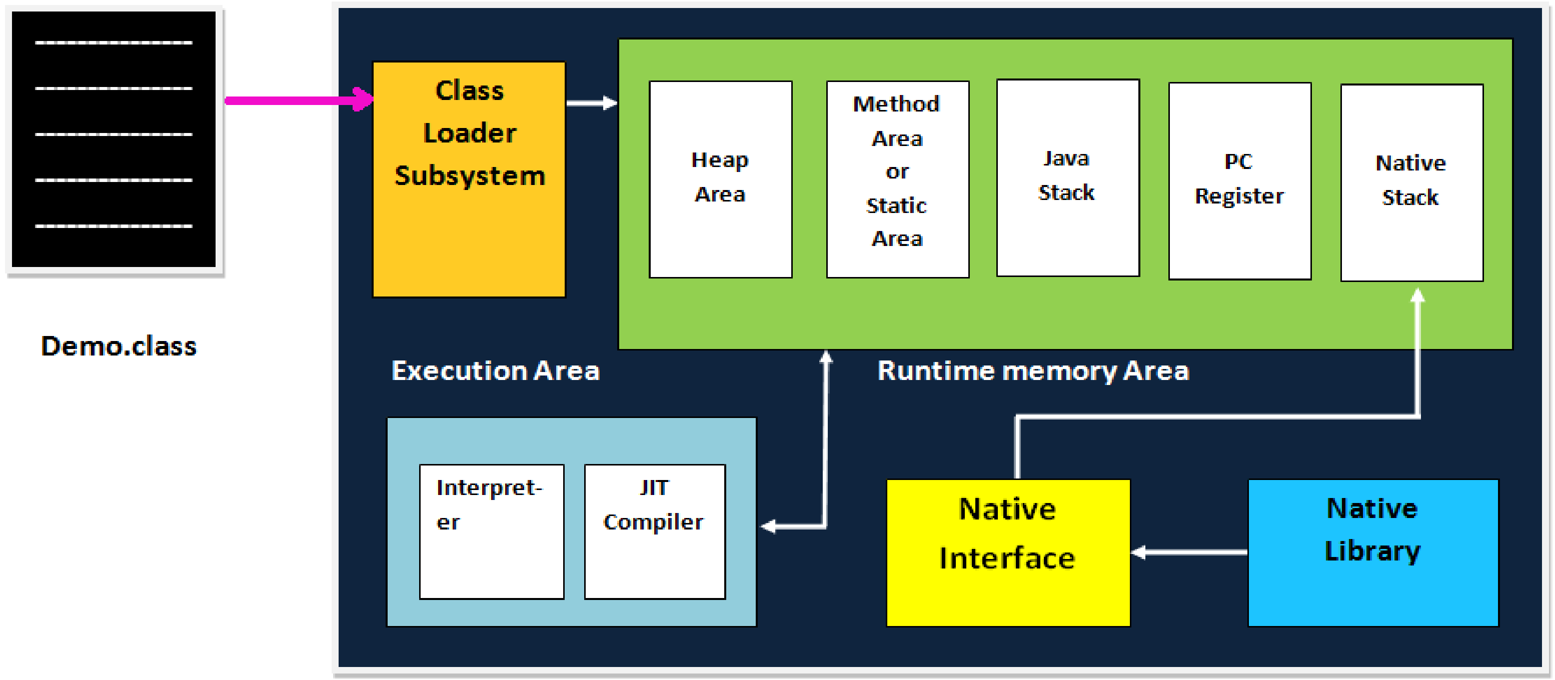
I am physically exists and contains Java Tools, JVM, JRE



I contains set of libraries and tools

JVM Architecture in Java

JVM (Java Virtual Machine) is a software. It is a specification that provides Runtime environment in which java bytecode can be executed.



Operation of JVM

- Allocating sufficient memory space for the class properties.
- Provides runtime environment in which java bytecode can be executed
- Converting byte code instruction into machine level instruction.

JVM is separately available for every Operating System while installing java software so that JVM is platform dependent.

Java is **platform Independent** but **JVM** is **platform dependent** because every operating system have different JVM which is installed along with JDK Software.

Class loader subsystem

Class loader subsystem will load the .class file into java stack and later sufficient memory will be allocated for all the properties of the java program into following five memory locations.

• Heap area	⇨	**In which object references will be stored**
• Method area	⇨	**In which static variables non-static and static method will be stored.**
• Java stack	⇨	**In which all the non-static variable of class will be stored and whose address referred by object reference.**
• PC register	⇨	**Which holds the address of next executable instruction that means that use the priority for the method in the execution process?**
• Native stack	⇨	**Native stack holds the instruction of native code (other than java code) native stack depends on native library. Native interface will access interface between native stack and native library.**

Execution Engine

Contains **Interpreter** and **JIT compiler**

$$\begin{tikzcd}\text{Java program} \arrow[r, "\text{Execution}", "\text{Interpreter}"'] & \text{Byte code} \arrow[r, "\text{JIT Compiler}"'] & \text{Machine level Instruction}\end{tikzcd}$$

JIT is the set of programs **developed by SUN Micro System** and added as a part of JVM, to **speed** up the **interpretation** phase.

Object and class in Java

Object is the physical as well as logical entity whereas class is the only logical entity. (i.e.) instance of a class

Class: Class is a blue print which is containing only list of variables and method and no memory is allocated for them. A class is a group of objects that has common properties.

Object and class in Java

A class contains:

- Data Member
- Method
- Constructor
- Block
- Class and Interface

Object and class in Java

Object: Object is a instance of class.

An object **has three characteristics:**

- State
- Behavior
- Identity

State: Represents data (value) of an object.

Behavior: Represents the behavior (functionality) of an object such as deposit, withdraw etc.

Identity: A unique ID used internally by the JVM to identify each object uniquely.

Object and class in Java

Examples

dog, cat, and cow are belong to **animal's class**.

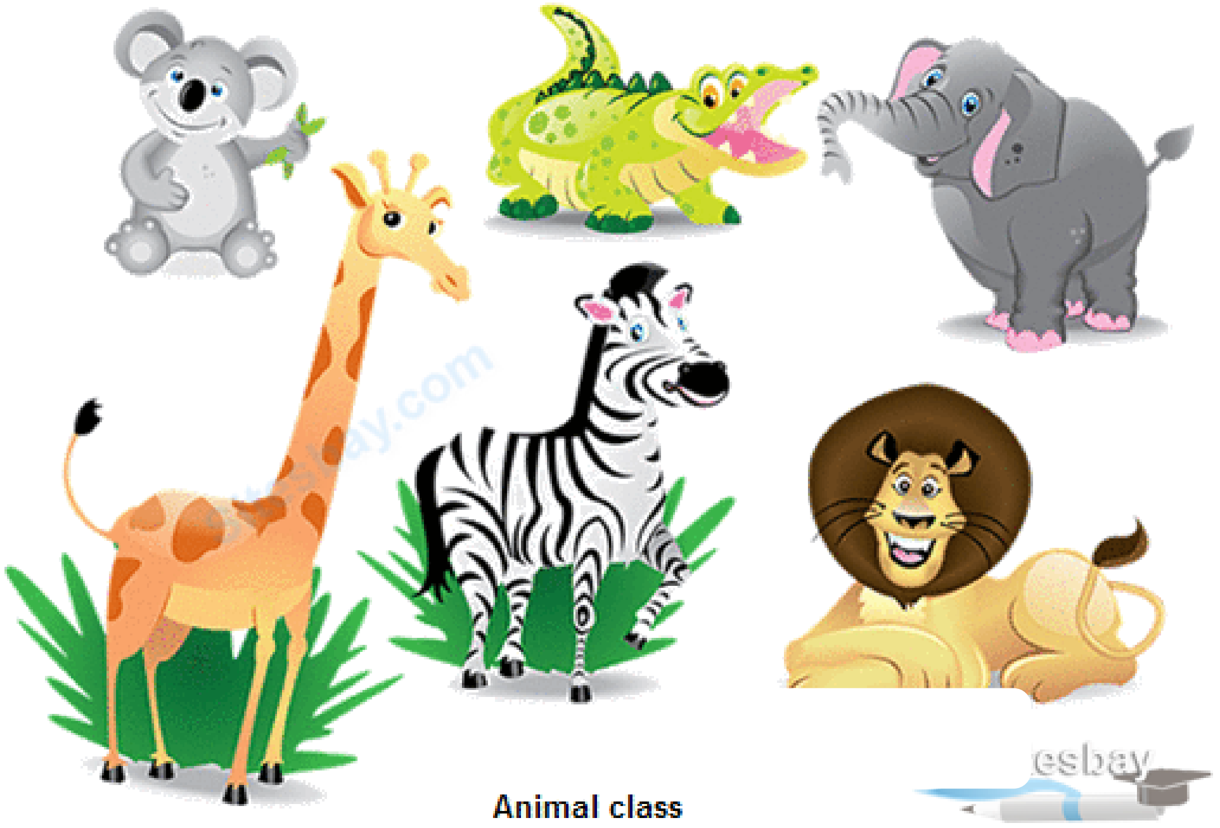
dog, cat, and cow – **Objects** of animal class

A dog (object properties) has

state:- color, name, height, age

behaviors:- barking, eating, and sleeping.

Object and class in Java



Animal class

Difference between Class and Object

	Class	Object
1	Class is a container with collection of variables and methods.	object is a instance of class
2	No memory is allocated at the time of declaration	Sufficient memory space will be allocated for all the variables of class at the time of declaration.
3	One class definition should exist only once in the program.	For one class multiple objects can be created.

Java Naming Convention

1. Every package name should exist a lower case letter.
2. First letter of every word of class name or interface name should exists in upper case.
3. Every constant value should exists in upper case letter. It is containing more than one word than it should be separated with underscore (-).
4. While declaring variable name, method, object reference the first letter of first word should exits in lower case but from the second words onward the first letter should exists in upper case.

Class

Syntax

```
class Class_Name
 {
   data member;
   method;
 }
```

Class

```
class Employee
{
  int eid; // data member (or instance variable)
  String ename; // data member (or instance variable)
  eid=101;
  ename= "Hitesh";
  public static void main(String args[])
   {
     Employee e=new Employee(); // Creating an object of class Employee
     System.out.println("Employee ID: "+e.eid);
     System.out.println("Name: "+e.ename);
    }
}
```

```
Employee ID: 101
Name: Hitesh
```

Wrapper classes

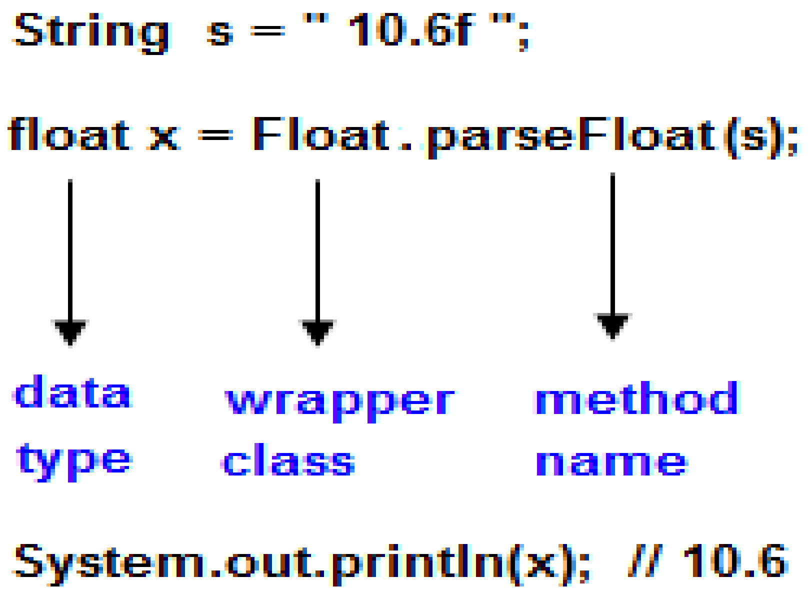
A **wrapper class** is a class that encapsulates (or "wraps") a primitive data type within an object. The purpose of wrapper class is to **convert numeric string data into numerical or fundamental data.**

Classes

```
class WraperDemo {

public static void main(String[] args) {

String s[] = {"10", "20"};

System.out.println("Sum before:"+ s[0] + s[1]); // 1020

int x=Integer.parseInt(s[0]); // convert String to Integer

int y=Integer.parseInt(s[1]); // convert String to Integer

int z=x+y; System.out.println("sum after: "+z); // 30 } }
```

```
Sum before: 1020

Sum after: 30
```

Wrapper Classes



```
String  s = " 10.6f ";

float x = Float.parseFloat(s);
```

```
System.out.println(x);  // 10.6
```

Wrapper Classes

Fundamental DataType	Wrapper CalssName	Conversion method from numeric string into fundamental or numeric value
byte	Byte	public static byte parseByte(String)
short	Short	public static short parseShort(String)
int	Integer	public static integer parseInt(String)
long	Long	public static long parseLong(String)
float	Float	public static float parseFloat(String)
double	Double	public static double parseDouble(String)
char	Character	provides useful methods for working with characters Character.isDigit('5’); Character.toUpperCase('b');
boolean	Boolean	public static boolean parseBoolean(String)

How to use wrapper class

All the wrapper class methods are static in nature so we need to call these method using **class.methodName().**

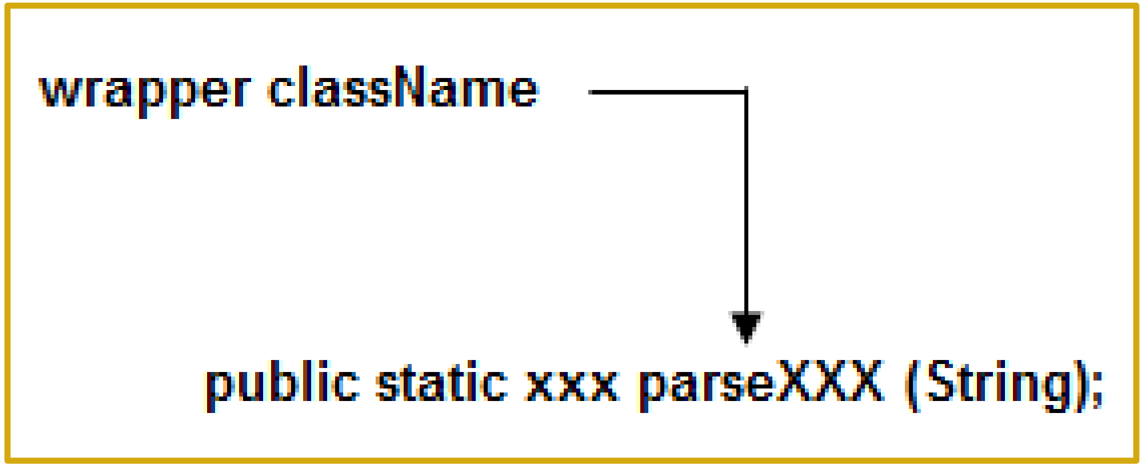
for Integer: int x=Integer.parseInt(String);

for float: float x=Float.parseFloat(String);

for double: double x=Double.parseDouble(String);

Each and every wrapper class contains the following generalized method for converting numeric String into fundamental values.

How to use wrapper class



Here xxx represents any fundamental data type.

Thank you