2-Day Workshop

Java Backend Development

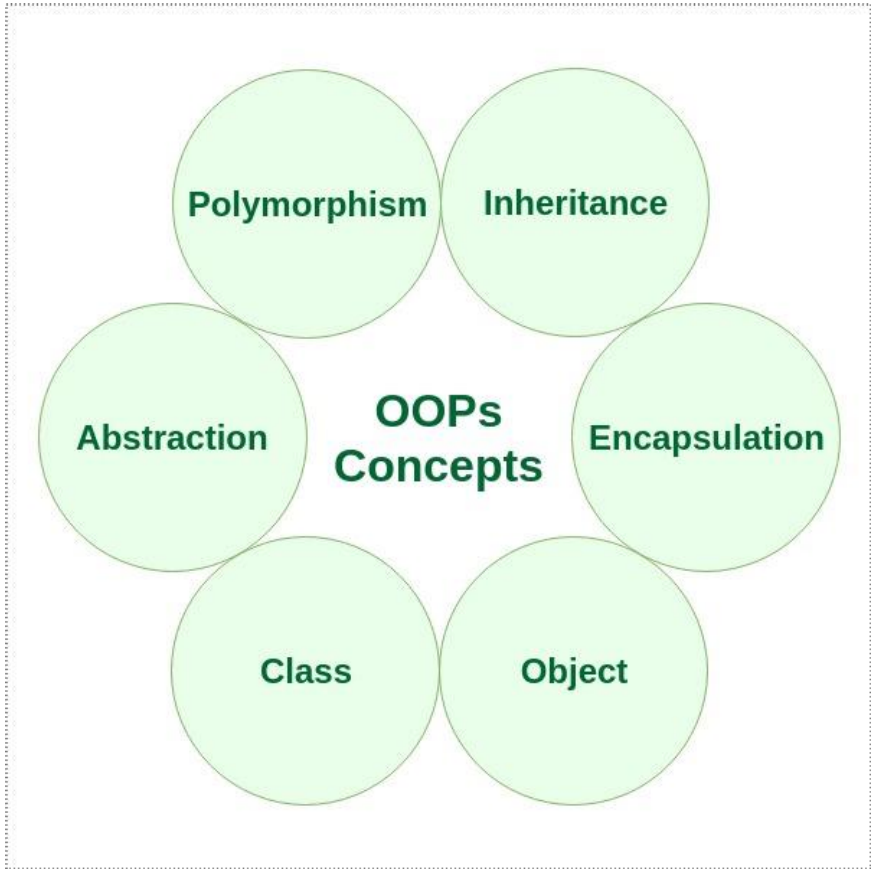
with

Chamandeep Singh
Course Mentor at GeeksforGeeks

Why Java ??

- **Java's Popularity and High Salary**
- **Java has a Large Community**
- **Java has Powerful Development Tools**
- **Java is Platform Independent**
- **Java has great Documentation Support**

Java OOPS


Polymorphism
Inheritance
Abstraction
OOPs Concepts
Encapsulation
Class
Object

• Class

A class is a user defined blueprint or prototype from which objects are created.
Class doesn't consume any space.

Example:-

```
public class Dog {
        String breed;              // Properties
        int age;
        String color;

        Dog() { }                  // Constructor

        void hungry() { }          // Methods
        void sleeping() { }
}
```

• Object

- An Object can be defined as an instance of a class.
- Every object takes space in memory and an object variable contains reference to space in memory.

Example:-

```
way 1:-     Dog dog = new Dog();

Way 2:-     Dog dog = new Dog(black);
```

• Inheritance

- When one object acquires all the properties and behaviors of a parent object, it is known as inheritance.
- It provides code reusability.
- It is used to achieve runtime polymorphism.
- Multiple inheritance is not allowed in Java.

Example:-

```
class A{}
class B extends A{}
```

Abstraction

- Hiding internal details and showing functionality is known as abstraction. For example phone call, we don't know the internal processing.
- In Java, we use abstract class and interface to achieve abstraction.

• Polymorphism

The word polymorphism means having many forms.

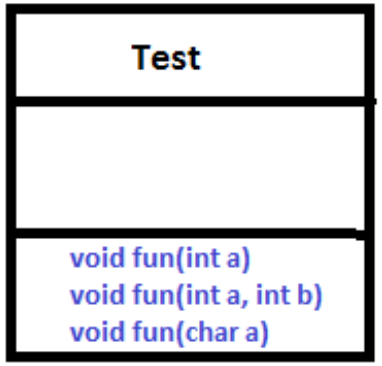
Example of polymorphism: A person at the same time can have different characteristic. Like a man at the same time is a father, a husband, an employee. So the same person posses different behavior in different situations. This is called polymorphism.

In Java polymorphism is mainly divided into two types:

•Compile time Polymorphism
•Runtime Polymorphism

• Polymorphism

Compile time polymorphism: Whenever an object is bound with their functionality at the compile-time, this is known as the compile-time polymorphism. The runtime polymorphism can be achieved by method overloading.

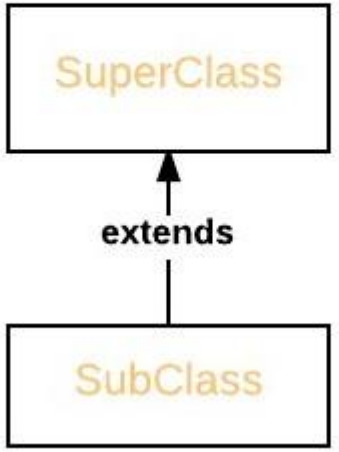


Overloading

Runtime polymorphism : Whenever an object is bound with the functionality at run time, this is known as runtime polymorphism. The runtime polymorphism can be achieved by method Overriding.

Upcasting

SuperClass obj = new SubClass

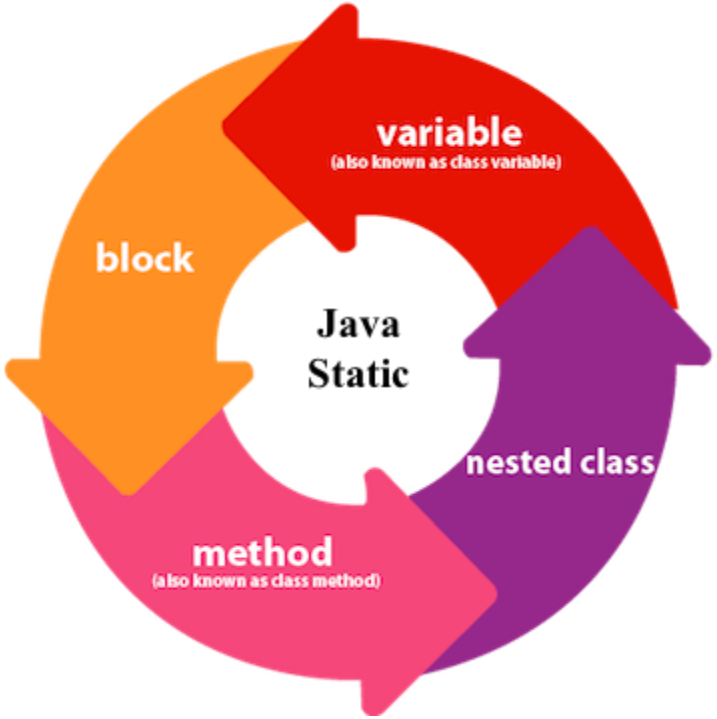


Encapsulation

- Binding (or wrapping) code and data together into a single unit are known as encapsulation. For example, a capsule, it is wrapped with different medicines.
- Java bean is the fully encapsulated class because all the data members are private.
- By making properties private we can encapsulate the data.

STATIC KEYWORD IN JAVA

The **static keyword** in Java is used for defining class level things.



https://www.geeksforgeeks.org/static-keyword-java/

• Static Blocks

- Static Block is called before anything (even before Main function).
- When there are two or more static blocks, then it is executed serially.
- As soon as class is loaded in JVM, static block is executed.

Syntax:-

```
Static{
        System.out.println("hey there !!");
}
```

Static Variables

- When a variable is declared as static, then a single copy of variable is created and shared among all objects at class level
- We can create static variables at class-level only.
- Static block and static variables are executed in order they are present in a program.

Static Methods

- The most common example of a static method is *main( )* method.

- Methods declared as static have several restrictions:
  - They can only directly call other static methods.
  - They can only directly access static data.
  - They cannot refer to this or super in any way.

Static Nested Classes

- We can not declare top-level class with a static modifier, but can declare nested Classes as static.
- Example :- Builder Design Pattern uses nested classes to create instance of outer class.

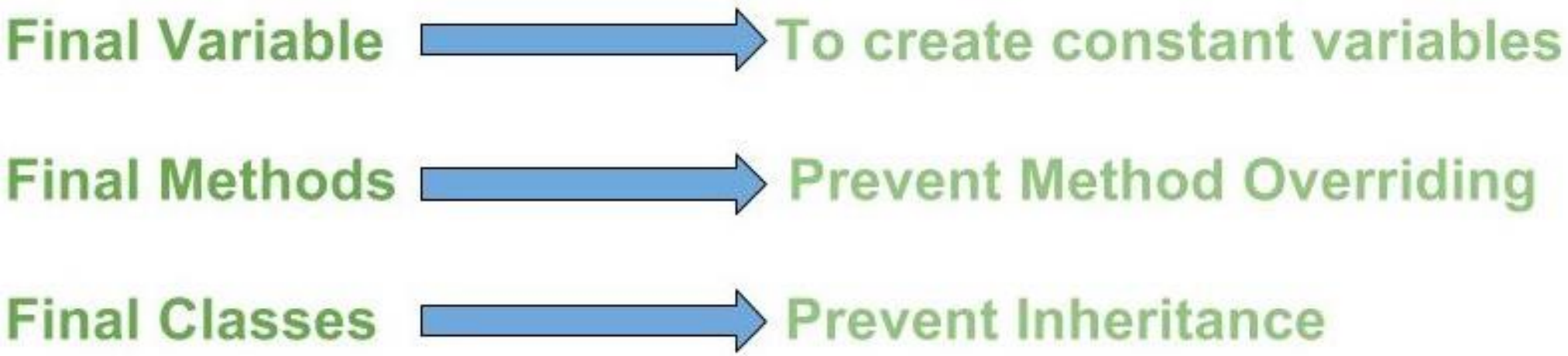
Singleton Design Pattern

- A singleton class is a class that can have only one object (an instance of the class) at a time.
- To design a singleton class:
  - Make constructor as private.
  - Write a static method that has return type object of this singleton class.

```java
class Singleton
{
    // static variable single_instance of type Singleton
    private static Singleton single_instance = null;

    // variable of type String
    public String s;

    // private constructor restricted to this class itself
    private Singleton() {
        s = "Hello I am a string part of Singleton class";
    }

    // static method to create instance of Singleton class
    public static Singleton getInstance() {
        if (single_instance == null)
            single_instance = new Singleton();

        return single_instance;
    }
}

// Driver Class
class Main {
    public static void main(String args[]) {
        Singleton x = Singleton.getInstance();
    }
}
```

FINAL KEYWORD IN JAVA

The **final keyword** in java is used to restrict the user. The java final keyword can be used in many context



Java Memory Management

Why Learn Java Memory Management?

- The automatic garbage collection doesn’t guarantee everything.
- If we don’t know how the memory management works, often we will end up amidst things that are not managed by JVM (Java Virtual Machine).
- There are some objects that aren’t eligible for the automatic garbage collection.

• JVM Stacks

•A stack is created at the same time when a thread is created and is used to store data and partial results which will be needed while returning value for method and performing dynamic linking.

• Stacks can either be of fixed or dynamic size. The size of a stack can be chosen independently when it is created.

• The memory for stack needs not to be contiguous.

• Heap

- It is a shared runtime data area and stores the actual object in a memory. It is instantiated during the virtual machine startup.
- This memory is allocated for all class instances and array. Heap can be of fixed or dynamic size depending upon the system’s configuration.
- JVM provides the user control to initialize or vary the size of heap as per the requirement. When a new keyword is used, object is assigned a space in heap, but the reference of the same exists onto the stack.
- There exists one and only one heap for a running JVM process.

• Garbage Collector

How Garbage Collector is Implemented ??

- Pause every thread
- Check live references.
- Mark them as Alive.
- Complete Scan of heap and remove all objects which are not alive.
- It copies the alive objects and moves to contiguous location In memory.

Is Java “passed by value” or “passed by reference” ??

WEB SERVICES ?

Services which are exposed to the internet for the programmatic access are known as Web Services.

Example:- Share on Facebook feature on games.

Q. What is difference between http://github.com and http://api.github.com

Types Of Web Services :-

SOAP

- It is maintained and Designed by Committee.
- It have Defined Set of Rules.
- Each soap webservice should follow these set of rules.

RESTFUL

- There is no committee to tell what's right or wrong.
- It don't have fixed any set of rules.
- It is a architectural style.
- Goal is to make services as much restful as possible.

Constraints of RESTFUL Webservices :-

- Client Server Model
- Statelessness
- Layered Architecture
- Cache ability
- Uniform Resource Identifier
- Code on demand

WHY SPRING BOOT ??

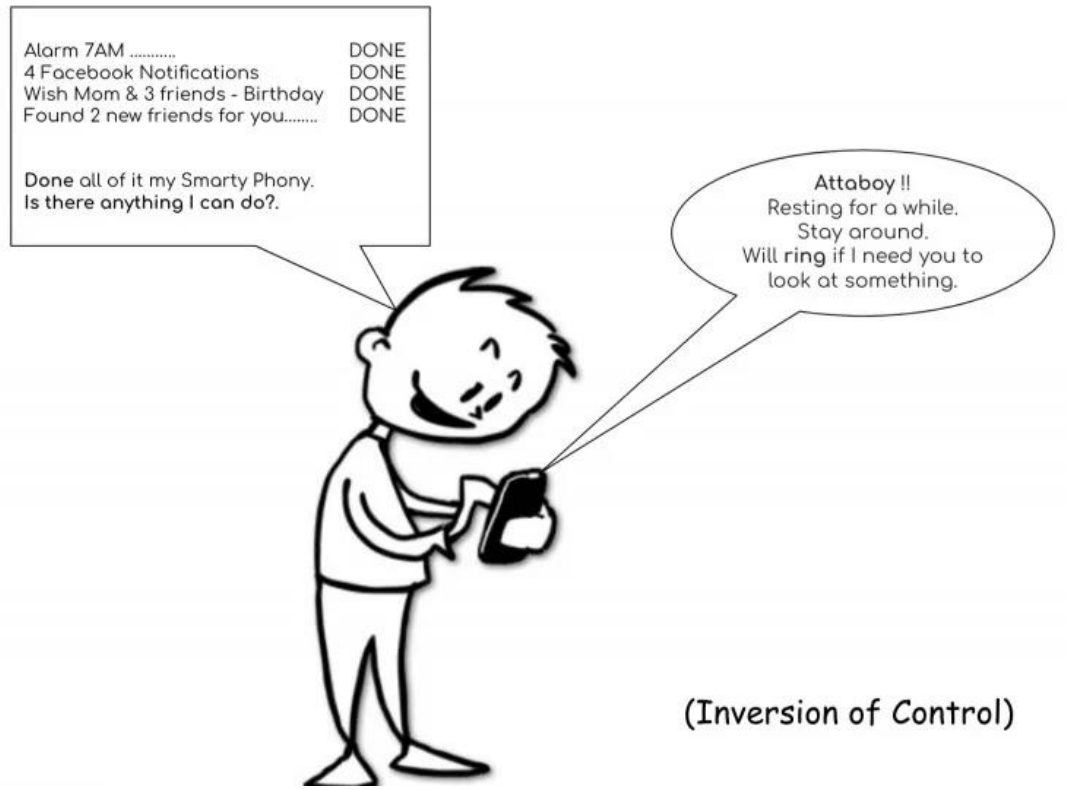
The following reasons why spring is being preferred for java application development :-

- Inversion of Control
- Aspect Oriented Programming
- Data Access
- Spring MVC

• Inversion Of Control

The inversion of Control is achieved by:-

- Application Context
- Dependency Injection



Data Access (Java Persistence API)

- JPA is a java specification which is used to manage relational data in application.
- JPA is not a framework. It a concept only.
- JPA is being implemented by ORM vendors such as Hibernate.