CHANDIGARH UNIVERSITY

Discover. Learn. Empower.

INSTITUTE : UIE

DEPARTMENT : CSE

Bachelor of Engineering (Computer Science & Engineering)

PROJECT BASED LEARNING IN JAVA

(20CST-319/20ITT-319)

TOPIC OF PRESENTATION:

Use of class and method in Java. Inheritance, Abstraction.

Lecture Objectives

In this lecture, we will discuss:

•Use of class and method in Java. Inheritance, Abstraction.

Class:

A class is a group of objects which have common properties. It is a template or blueprint from which objects are created. It is a logical entity. It can't be physical.

A class in Java can contain:

- Fields
- Methods
- Constructors
- Blocks (Instance Initialization Block (IIB), Static Initialization Block (SIB))
- Nested class and interface

Static Class:

- You cannot use the static keyword with a class unless it is an inner class.
- A static inner class is a nested class which is a static member of the outer class.
- It can be accessed without instantiating the outer class, using other static members.
- Just like static members, a static nested class has not have access to the instance variables and methods of the outer class.

Can a class be static in Java?

The answer is Yes, some classes can be made static in Java. Java supports

- Static Instance Variables
- Static Methods,
- Static Block
- Static Classes

Java Inner Classes:

- In Java, it is also possible to nest classes (a class within a class). The purpose of nested classes is to group classes that belong together, which makes your code more readable and maintainable.
- To access the inner class, create an object of the outer class, and then create an object of the inner class.

Private Inner Class:

- Unlike a "regular" class, an inner class can be private or protected. If you don't want outside objects to access the inner class, declare the class as private.

Static Inner Class:

- An inner class can also be static, which means that you can access it without creating an object of the outer class

Access Outer Class From Inner Class:

- One advantage of inner classes, is that they can access attributes and methods of the outer class

Inheritance in Java

Inheritance in Java is a mechanism in which one object acquires all the properties and behaviors of a parent object. It is an important part of OOPs (Object Oriented programming system).

Inheritance represents the **IS-A relationship** which is also known as a *parent-child* relationship.

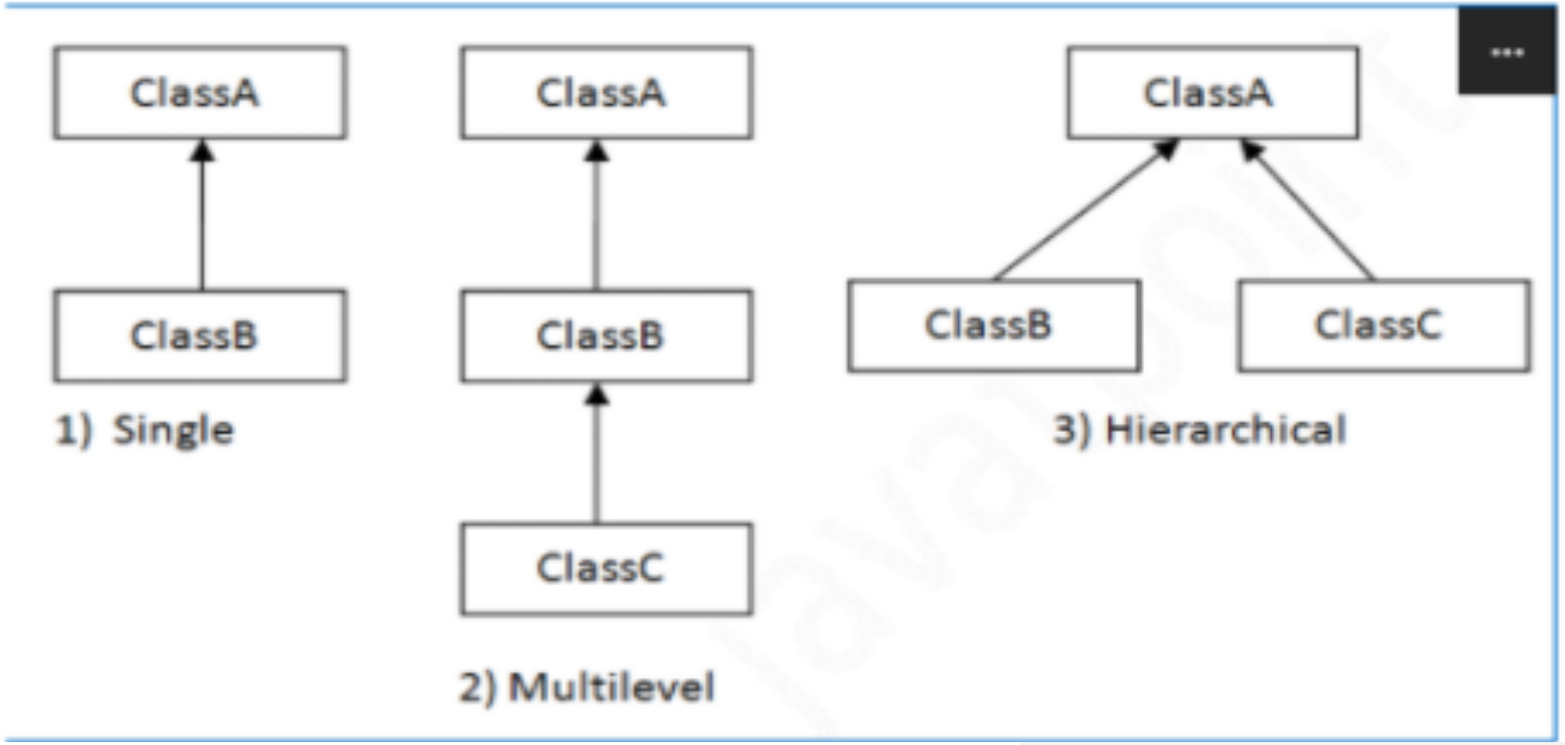
Why use inheritance in java:

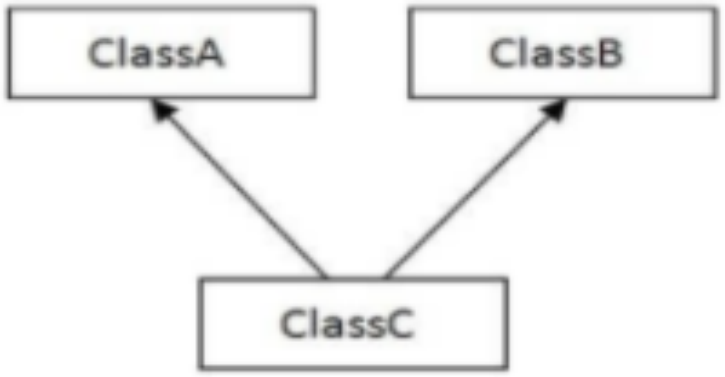
- For Method Overriding (so runtime polymorphism can be achieved).
- For Code Reusability.

Terms used in Inheritance:

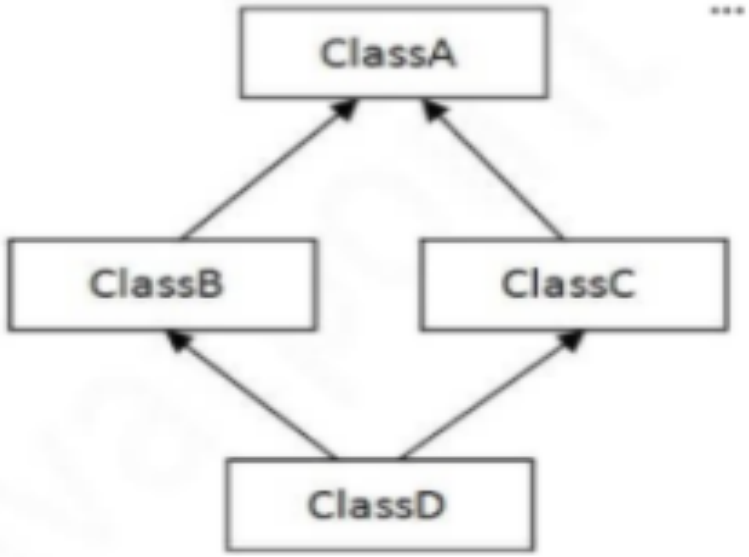
- Class
- Sub Class/Child Class
- Super Class/Parent Class
- Reusability

Types of inheritance in java





4) Multiple



5) Hybrid

Abstraction in Java

Abstraction is a process of hiding the implementation details and showing only functionality to the user.

Abstraction lets you focus on what the object does instead of how it does it.

Ways to achieve Abstraction

There are two ways to achieve abstraction in java

- Abstract class (0 to 100%)
- Interface (100%)

Abstract Classes

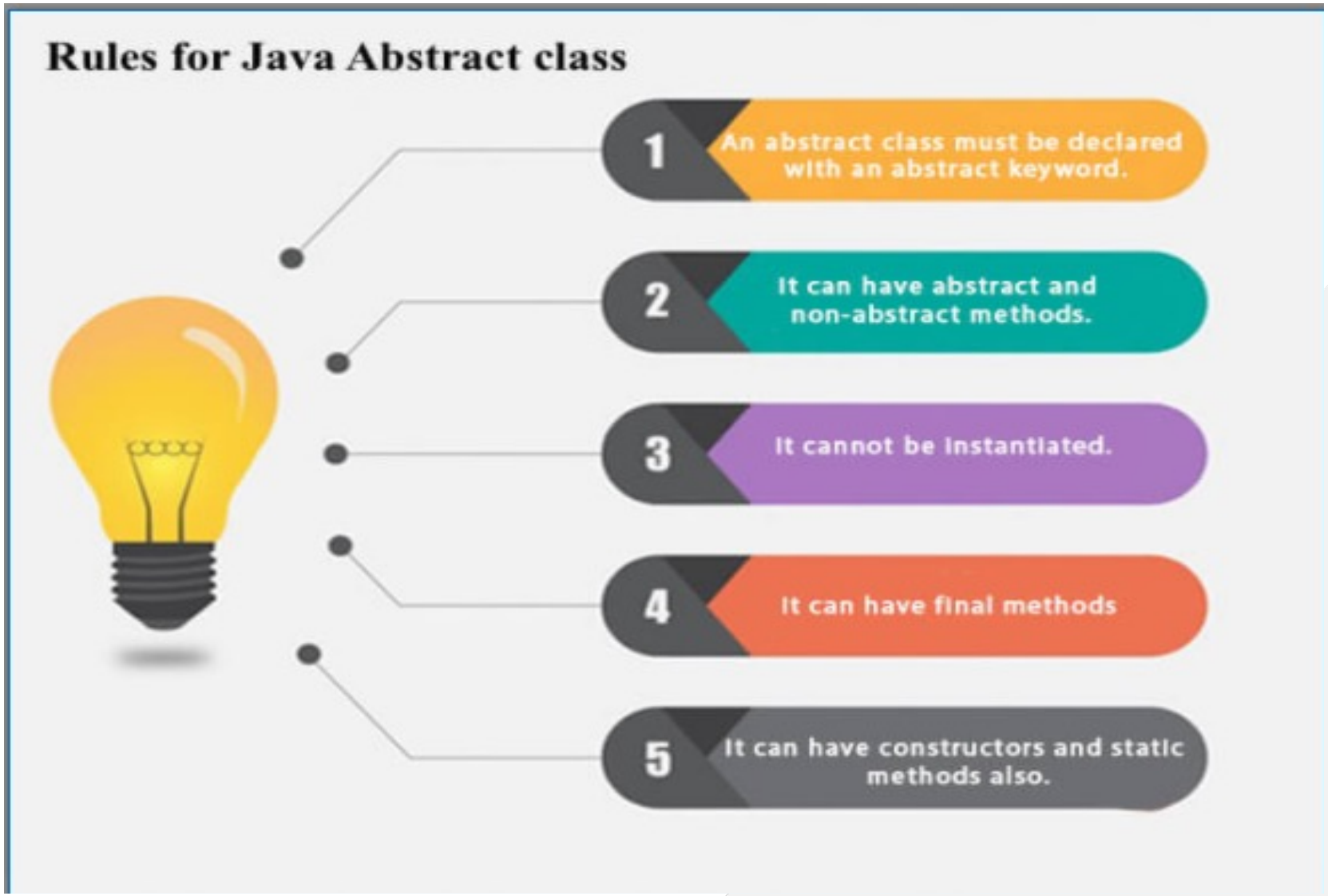
A class that is declared with the abstract keyword is known as an abstract class in Java. It can have abstract and non-abstract methods (method with the body).

Abstract class in Java

- A class that is declared as abstract is known as an **abstract class**. It can have abstract and non-abstract methods. It needs to be extended and its method implemented. It cannot be instantiated.

Points to Remember

- An abstract class must be declared with an abstract keyword.
- It can have abstract and non-abstract methods.
- It cannot be instantiated.
- It can have constructors and static methods also.
- It can have final methods that will force the subclass not to change the body of the method.


Rules for Java Abstract class
1
An abstract class must be declared with an abstract keyword.
2
It can have abstract and non-abstract methods.
3
It cannot be instantiated.
4
It can have final methods
5
It can have constructors and static methods also.

QUIZ:

1. Order of execution of constructors in Java Inheritance is

a. Super to sub class
b. Sub to super class
c. Random order
d. None

2. If a class inheriting an abstract class does not define all of its function then it will be known as?

a) Abstract
b) A simple class
c) Static class
d) None of the mentioned

Summary:

In this session, you were able to :

- Learn about Use of class and method in Java. Inheritance, Abstraction.

References:

Books:

1. Balaguruswamy, *Java.*
2. A Primer, E.Balaguruswamy, *Programming with Java,* Tata McGraw Hill Companies
3. John P. Flynt Thomson, *Java Programming.*

Video Lectures :

https://www.youtube.com/watch?v=okruEgWGVGU
https://www.youtube.com/watch?v=V7yVbG9_xkM
https://www.youtube.com/watch?v=RcIsb9iFKH8
https://www.youtube.com/watch?v=p_4Dyfplqkw

Reference Links:

https://www.w3schools.com/java/java_inner_classes.asp
https://www.geeksforgeeks.org/inner-class-java/
https://www.tutorialspoint.com/java/java_innerclasses.html
https://www.w3schools.com/java/java_abstract.asp
https://www.javatpoint.com/abstract-class-in-java

THANK YOU