GLS UNIVERSITY

SEM – IV
0301401 - CORE JAVA

Unit-3

-
- **Inheritance**
  - Introduction
  - Types of Inheritance
  - Deriving classes using extend keyword
  - Overriding of Methods
  - Super Keyword
  - Final keyword
  - Abstract class

- **Interfaces**
  - Introduction
  - Variables in interface
  - Extending interface
  - Interface v/s Abstract class

Unit – 3
Inheritance in Java

Introduction

- Inheritance in java is a mechanism in which one object acquires all the properties and behaviors of parent object.
- The idea behind inheritance in java is that you can create new classes that are built upon existing classes.
- When you inherit from an existing class, you can reuse methods and fields of parent class, and you can add new methods and fields also.
- Inheritance represents the **IS-A relationship**, also known as parent-child relationship.

Introduction

- Why use inheritance in java
  - For Method Overriding
  - For Code Reusability.

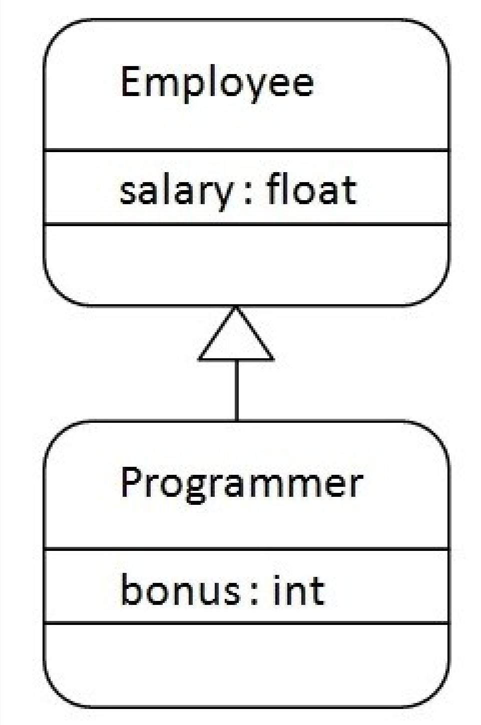
Syntax of Java Inheritance

```
class Subclass-name extends Superclass-name
{
   //methods and fields
}
```

The extends keyword indicates that you are making a new class that derives from an existing class. The meaning of "extends" is to increase the functionality.

In the terminology of Java, a class which is inherited is called parent or super class and the new class is called child or subclass.

Introduction

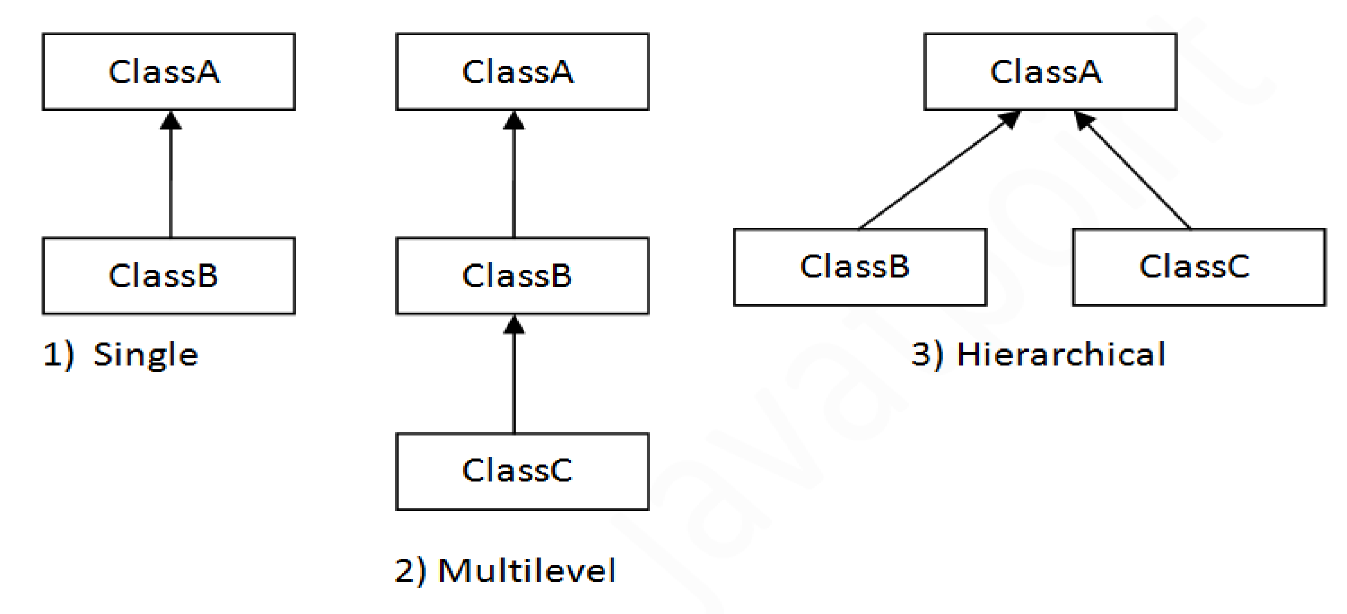


As displayed in the above figure, Programmer is the subclass and Employee is the superclass. Relationship between two classes is Programmer IS-A Employee.

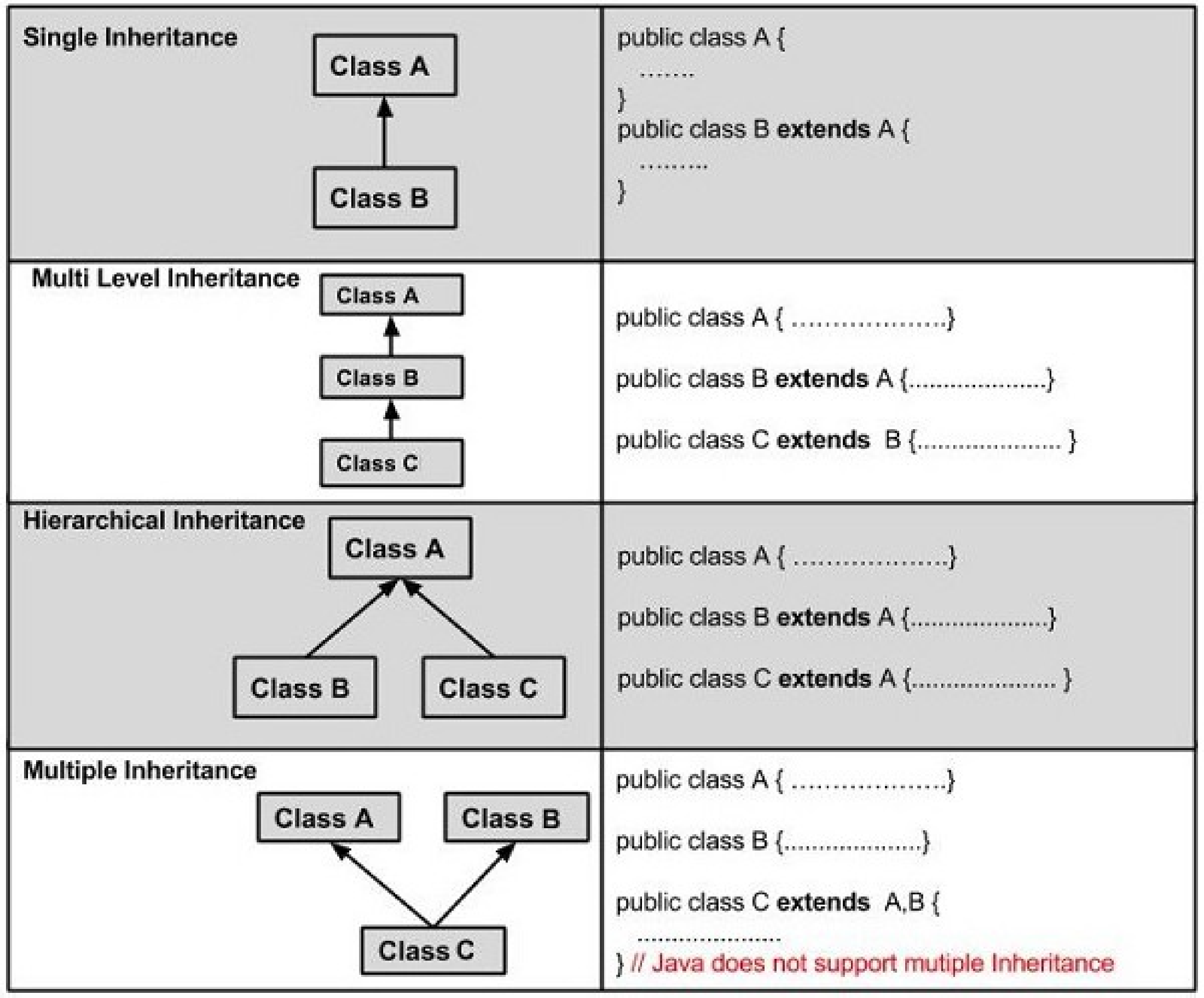
It means that Programmer is a type of Employee.

Types of Inheritance

- On the basis of class, there can be three types of inheritance in java: single, multilevel and hierarchical.
- In java programming, multiple and hybrid inheritance is supported through interface only.
- Multiple inheritance is not supported in java through class.



Types of Inheritance



Single Inheritance
Class A
Class B
	public class A {
	…….
	}
	public class B **extends** A {
	………
	}

Multi Level Inheritance
Class A
Class B
Class C
	public class A { ……………….}
	public class B **extends** A {………………..}
	public class C **extends** B {………………… }

Hierarchical Inheritance
Class A
Class B Class C
	public class A { ……………….}
	public class B **extends** A {………………..}
	public class C **extends** A {………………… }

Multiple Inheritance
Class A Class B
Class C
	public class A { ……………….}
	public class B {………………..}
	public class C **extends** A,B {
	…………………
	} // Java does not support mutiple Inheritance

Why multiple inheritance is not supported in java?

- To reduce the complexity and simplify the language, multiple inheritance is not supported in java.
- Consider a scenario where A, B and C are three classes. The C class inherits A and B classes.
- If A and B classes have same method and you call it from child class object, there will be ambiguity to call method of A or B class.
- Since compile time errors are better than runtime errors, java renders compile time error if you inherit 2 classes.
- So whether you have same method or different, there will be compile time error now.

Method Overriding

- If subclass (child class) has the same method as declared in the parent class, it is known as method overriding in java.
- In other words, If subclass provides the specific implementation of the method that has been provided by one of its parent class, it is known as method overriding.

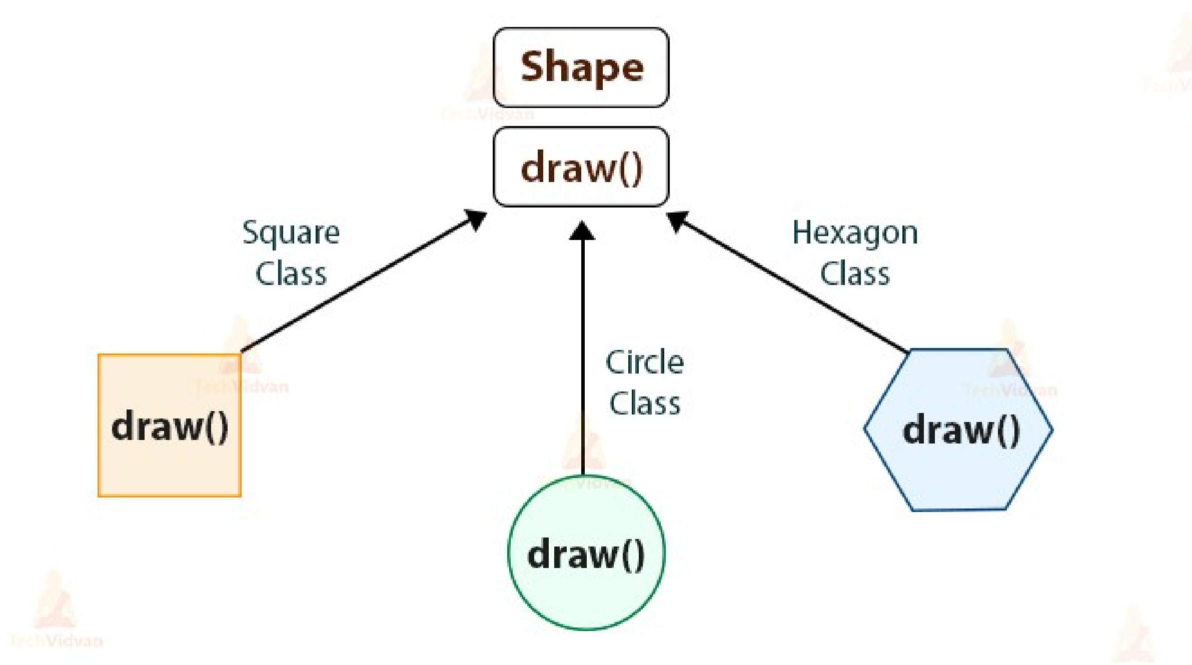
Usage of Java Method Overriding

- Method overriding is used to provide specific implementation of a method that is already provided by its super class.
- Method overriding is used for runtime polymorphism

Example of Method Overriding

Rules for Java Method Overriding

- Method must have same name as in the parent class
- Method must have same parameter as in the parent class.
- Must be IS-A relationship (inheritance).



Super keyword in java

- The super keyword in java is a reference variable which is used to refer immediate parent class object.
- Whenever you create the instance of subclass, an instance of parent class is created implicitly which is referred by super reference variable.

Usage of java super Keyword :

- super can be used to refer immediate parent class instance variable.
- super can be used to invoke immediate parent class method.
- super() can be used to invoke immediate parent class constructor.

super() is added in each class constructor automatically by compiler if there is no super() or this().