UNITED

GROUP OF INSTITUTIONS

# Web Technology (KCS-602) Unit 1

Prepared By

Abhishek Kesharwani

**Assistant Professor,UCER Naini,Allahabad**

# Lecture 7

- Inheritance

# Inheritance in Java

- **Inheritance in java** is a mechanism in which one object acquires all the properties and behaviours of parent object.
- The idea behind inheritance in java is that you can create new classes that are built upon existing classes.
- Inheritance represents the **IS-A relationship**, also known as *parent-child* relationship.

## Why use inheritance in java

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

## Syntax of Java Inheritance

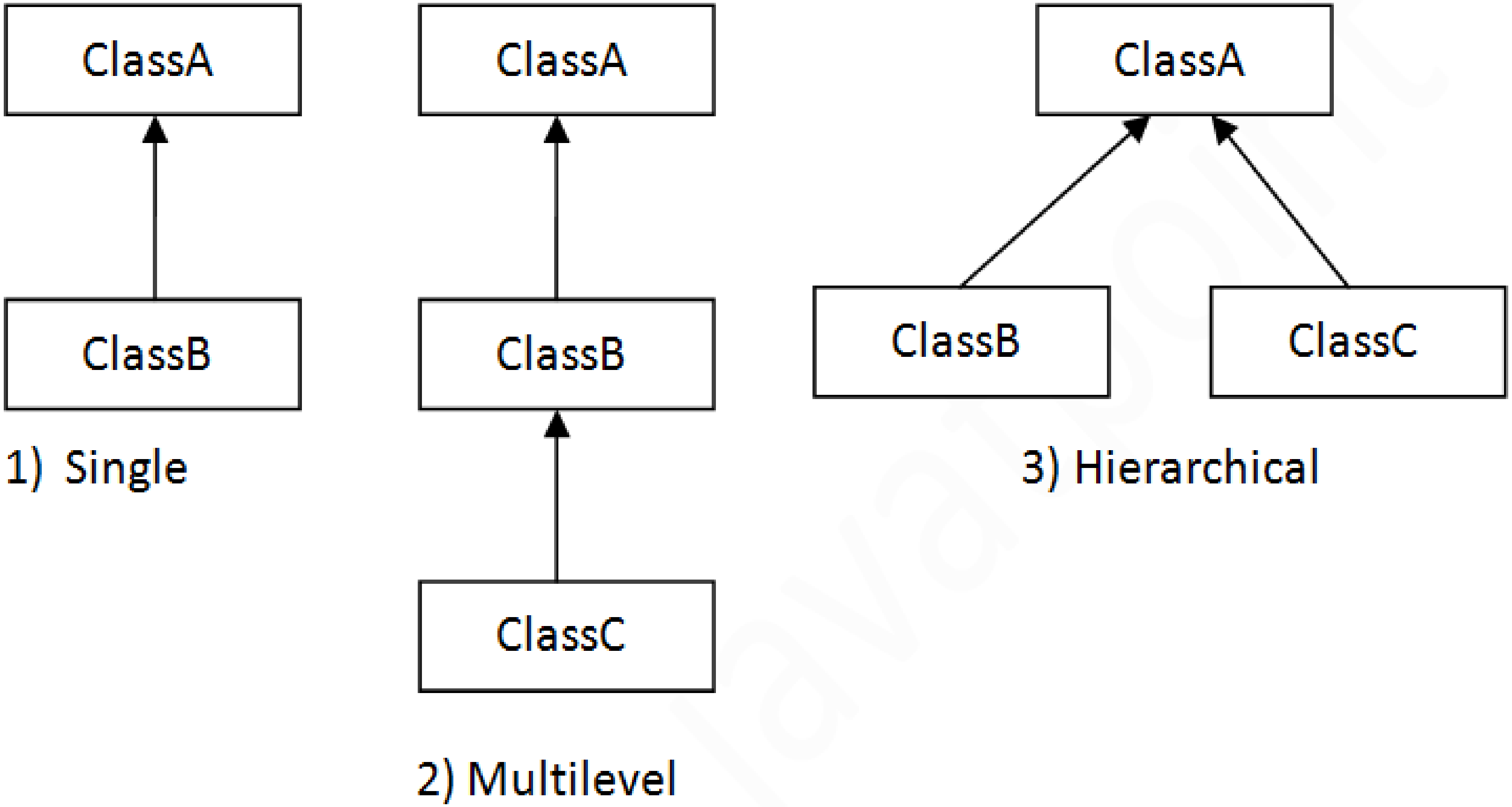
```
class Subclass-name extends Superclass-name
{
  //methods and fields
}
```

The **extends keyword** indicates that you are making a new class that derives from an existing class.

```
class Employee{
 float salary=40000;
}
class Programmer extends Employee{
 int bonus=10000;
 public static void main(String args[]){
   Programmer p=new Programmer();
   System.out.println("Programmer salary is:"+p.salary);
   System.out.println("Bonus of Programmer is:"+p.bonus)
    ;
}
}
```

```
Programmer salary is:40000.0
Bonus of programmer is:10000
```

# Types of inheritance in java



ClassA

ClassB

1) Single

ClassA

ClassB

ClassC

2) Multilevel

ClassA

ClassB

ClassC

3) Hierarchical

# 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.

```
class A{
void msg(){System.out.println("Hello");}
}
class B{
void msg(){System.out.println("Welcome");}
}
class C extends A,B{//suppose if it were

 Public Static void main(String args[]){
   C obj=new C();
   obj.msg();//Now which msg() method would be invoked?
}
}
```

Compile Time Error

# Java Method Overloading

```
class OverloadingExample
{
static int add(int a,int b)
{
return a+b;
}
static int add(int a,int b,int c)
{
return a+b+c;
}
}
```

# Method Overriding in Java

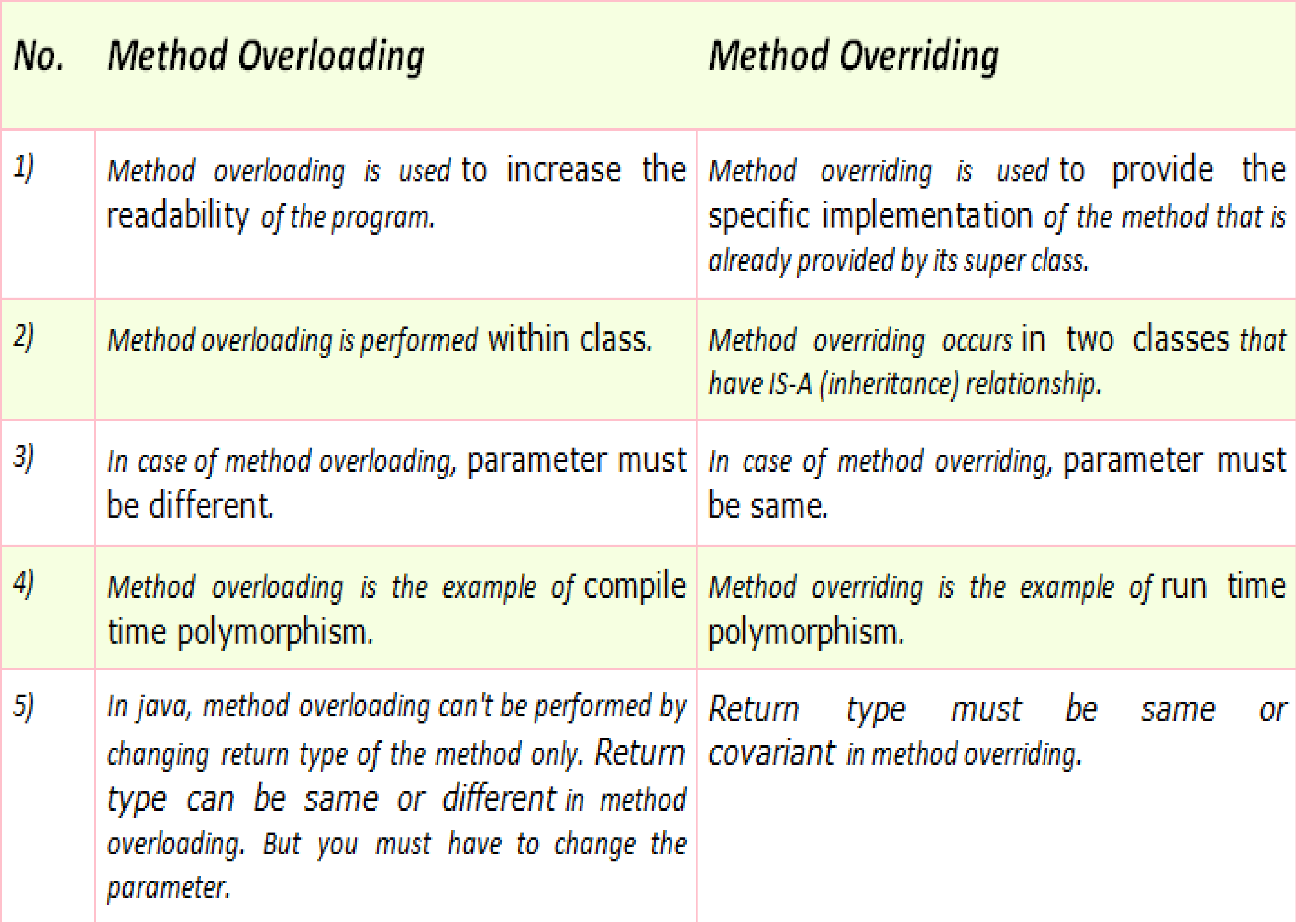
- If subclass (child class) has the same method as declared in the parent class, it is known as **method overriding in java**.
- Method overriding is used to provide specific implementation of a method that is already provided by its super class.
- Method overriding is used for runtime polymorphism.

# 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).

```
class Vehicle{
 void run(){System.out.println("Vehicle is running");}
}
class Bike extends Vehicle{  void run(){System.out.println(“Bike  is running");}


 public static void main(String args[]){
 Bike obj = new Bike();
 obj.run();
 }
}
```



| No. | Method Overloading | Method Overriding |
|---|---|---|
| 1) | Method overloading is used to increase the readability of the program. | Method overriding is used to provide the specific implementation of the method that is already provided by its super class. |
| 2) | Method overloading is performed within class. | Method overriding occurs in two classes that have IS-A (inheritance) relationship. |
| 3) | In case of method overloading, parameter must be different. | In case of method overriding, parameter must be same. |
| 4) | Method overloading is the example of compile time polymorphism. | Method overriding is the example of run time polymorphism. |
| 5) | In java, method overloading can't be performed by changing return type of the method only. Return type can be same or different in method overloading. But you must have to change the parameter. | Return type must be same or covariant in method overriding. |

# Question Asked in Interview

- **How is Inheritance implemented/achieved in Java?[TCS]**
- **Which class in Java is superclass of every other class? [Infosys]**
- **Can we extend (inherit) final class?**
- Ans: No, a class declared with final keyword cannot be inherited.
- **Can a final method be overridden?**
- Ans: No, a final method cannot be overridden.