**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**.

In other words, If a subclass provides the specific implementation of the method that has been declared by one of its parent class, it is known as method overriding.

Usage of Java Method Overriding

* Method overriding is used to provide the specific implementation of a method which is already provided by its superclass.
* Method overriding is used for runtime polymorphism

Rules for Java Method Overriding

1. The method must have the same name as in the parent class
2. The method must have the same parameter as in the parent class.
3. There must be an IS-A relationship (inheritance).

**package** com.ecstatic.aniket;

**class** RBI1 {

// Rules for Method overriding

// method name must be same

// parameter must be same

// ther must be IS-A relationship (Inheritance)

**public** **double** rateOfInterestPL() {

**return** 8.2; // 8.2 is rate of Interest

}

}

**class** HDFC1 **extends** RBI1{

**public** **double** rateOfInterestPL() {

**return** 8.5;

}

}

**class** ICICI1 **extends** RBI1{

**public** **double** rateOfInterestPL() {

**return** 9.5;

}

}

**class** BOM1 **extends** RBI1{

**public** **double** rateOfInterestPL() {

**return** 8.6;

}

}

**public** **class** PolyMorphismExample2 {

**public** **static** **void** main(String[] args) {

// Dynamic Method Dispatch

// Upcasting

// we have created the object of child class with parentr reference

RBI1 hdfc = **new** HDFC1(); // hdfc object is also called parent reference i.e. Reference of RBI

System.***out***.println("Rate of Interest of HDFC is :" +hdfc.rateOfInterestPL());

RBI1 icici = **new** ICICI1();

System.***out***.println("Rate of Interest of ICICI is :" +icici.rateOfInterestPL());

RBI1 bom = **new** BOM1();

System.***out***.println("Rate of Interest of BOM is :" +bom.rateOfInterestPL());

}

}

**There are many differences between method overloading and method overriding in java. A list of differences between method overloading and method overriding are given below:**

|  |  |  |
| --- | --- | --- |
| **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 overri |