



RAMA
UNIVERSITY

www.ramauniversity.ac.in

FACULTY OF Engineering &
Technology

Method Overriding

1. If subclass (child class) has the same method as declared in the parent class, it is known as **method overriding in Java**
2. In overriding parent child relationship must be needed

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).

```
class car{
    void run(){System.out.println("car is running");}
}
class Bike extends Vehicle{
    void run(){System.out.println("bike is flying now");}
}
public static void main(String args[]){
    Bike obj = new Bike();
    obj.run();
}
}
```

Here run method of bike class will run.

super Keyword

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

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

Final Keyword

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

1. variable
2. method
3. Class

If a class is declared as final then it can not be inherited, same if a method is declared as final then it can not be overridden in subclass

this keyword

1. this is a **reference variable** that refers to the current object.
2. In case of constructor it will call the constructor of the same class
3. this can be used to refer current class instance variable.
4. this can be used to invoke current class method (implicitly)
5. this() can be used to invoke current class constructor.
6. this can be passed as an argument in the method call.
7. this can be passed as argument in the constructor call.
8. this can be used to return the current class instance from the method.

```
class Student{  
int rollno;  
String name;  
Student(int rollno,String name){  
this.rollno=rollno; //here this refer to current object's rollno  
this.name=name; //here this refer to current object's name  
}  
void display(){System.out.println(rollno+" "+name);}  
}
```