

Runtime polymorphism

Runtime polymorphism is a process in which a call to an overridden method is resolved at runtime rather than compile-time.

In this process, an overridden method is called through the reference variable of a superclass. The determination of the method to be called is based on the object being referred to by the reference variable.

Example :

```
class Bike{  
    void run(){System.out.println("running");}  
}  
class Splender extends Bike{  
    void run(){System.out.println("running safely with 60km");}  
  
    public static void main(String args[]){  
        Bike b = new Splender();//upcasting  
        b.run();  
    }  
}
```

In this example, we are creating two classes Bike and Splendar. Splendar class extends Bike class and overrides its run()

method. We are calling the run method by the reference variable of Parent class. Since it refers to the subclass object and subclass method overrides the Parent class method, subclass method is invoked at runtime.

Since method invocation is determined by the JVM not compiler, it is known as runtime polymorphism.