Polymorphism in Java

Polymorphism is the capability of a method to do different things based on the object that it is acting upon. In other words, polymorphism allows you define one interface and have multiple implementations.

* It is a feature that allows one interface to be used for a general class of actions.
* An operation may exhibit different behavior in different instances.
* The behavior depends on the types of data used in the operation.
* Polymorphism is extensively used in implementing inheritance.

# Types of polymorphism in java

## Runtime Polymorphism (or Dynamic polymorphism)

## Compile time Polymorphism (or Static polymorphism)

Compile time polymorphism is nothing but the method overloading in java. In simple terms we can say that a class can have more than one method with same name but with different number of arguments or different types of arguments or both.

Suppose the class has three variance of method A or we can say method A is polymorphic in nature since it is having three different forms. In such scenario, compiler is able to figure out the method call at compile-time that’s the reason it is known as compile time polymorphism.

Following concepts demonstrate different types of polymorphism in java.  
[**Method Overloading**](http://beginnersbook.com/2013/05/method-overloading/)

Runtime Polymorphism (or Dynamic polymorphism

[**Method Overriding**](http://beginnersbook.com/2014/01/method-overriding-in-java-with-example/) (We will discuss it later)

**Method Overloading:**  
In Java, it is possible to define two or more methods of same name in a class, provided that there argument list or parameters are different. This concept is known as Method Overloading.