

## Polymorphism

**Q 1:** How many types polymorphism in Java ?

**Ans :**

1) Static Polymorphism also known as compile time polymorphism

2) Dynamic Polymorphism also known as runtime polymorphism

Source : <https://beginnersbook.com/2013/04/runtime-compile-time-polymorphism/>

**Types :**

1) Static binding/Compile-Time binding/Early binding/Method overloading.(in same class)

2) Dynamic binding/Run-Time binding/Late binding/Method overriding.(in different classes)

**Source :** <https://stackoverflow.com/questions/20783266/what-is-the-difference-between-dynamic-and-static-polymorphism-in-java>

**Q 2 :** What is Dynamic Method Dispatch ?

**Ans :** Dynamic method dispatch is the mechanism by which a call to an overridden method is resolved at run time, rather than compile time.

Source: <https://www.geeksforgeeks.org/dynamic-method-dispatch-runtime-polymorphism-java/>

**Q 3 :** What is Covariance and Contravariance ?

**Ans :** Covariance and contravariance determine compatibility based on types. In either case, variance is a directed relation. *Covariance* can be translated as “different in the same direction,” or *with-different*, whereas *contravariance* means “different in the opposite direction,” or *against-different*. Covariant and contravariant types are not the same, but there is a correlation between them. The names imply the direction of the correlation. So, *covariance* means that the compatibility of two types implies the compatibility of the types dependent on them.

Source : <http://programmersbhawa.com/covariance-contravariance-java/>

**Q 4:** What is the difference between Static and Dynamic Binding .?

**Ans :** 1) Static binding is done during compile-time while dynamic binding is done during run- time.

2) private, final and static methods and variables uses static binding and bonded by compiler while overridden methods are bonded during runtime based upon type of runtime object

Source : <https://www.geeksforgeeks.org/dynamic-method-dispatch-runtime-polymorphism-java/>

**Q 5 :** Define rules of Overriding .

**Ans :**

Rule #1: Only inherited methods can be overridden.

Rule #2: Final and static methods cannot be overridden.

Rule #3: The overriding method must have same argument list.

Rule #4: The overriding method must have same return type (or subtype).

Rule #5: The overriding method must not have more restrictive access modifier.

Rule #6: The overriding method must not throw new or broader checked exceptions.

Rule #7: Use the super keyword to invoke the overridden method from a subclass.

Rule #8: Constructors cannot be overridden.

Rule #9: Abstract methods must be overridden by the first concrete (non-abstract) subclass.

Rule #10: A static method in a subclass may hide another static one in a superclass, and that's called hiding.

Rule #11: The synchronized modifier has no effect on the rules of overriding.

Rule #12: The strictfp modifier has no effect on the rules of overriding.

Source : <http://www.codejava.net/java-core/the-java-language/12-rules-of-overriding-in-java-you-should-know>

**Q 6:** Define use of Overriding .

**Ans :** Functions with the same definition have different behavior in different scopes.

Used by a child class to change behavior it inherited from its parent.

Source : <https://www.careercup.com/question?id=1874672>

**Q 7 :**Describe Advantage and Disadvantage of Overriding .

**Ans :**

**Advantages :**

- 1)Flexibility:Overloaded methods give programmers the flexibility to call a similar method for different types of data.
- 2) Constructors : Overloading is also used on constructors to create new objects given different amounts of data.

### Disadvantages :

- 1) Ambiguous References : Overloaded methods must use different numbers or types of arguments to avoid ambiguity
- 2) Return Types : You must define a return type for each overloaded method. Methods can have different return types

Source : <https://www.techwalla.com/articles/the-advantages-disadvantages-of-the-overloading-method-in-java>

**Q 8 :** Describe difference between Runtime and Compile time Polymorphism.

**Ans :**

### Compile time Polymorphism

- 1) In Compile time Polymorphism, call is resolved by the **compiler**.
- 2) It is also known as **Static binding**, **Early binding** and **overloading** as well.
- 3) **Overloading** is compile time polymorphism where more than one methods share the same name with different parameters or signature and different return type.
- 4) It is achieved by **function** overloading and **operator** overloading.
- 5) It provides **fast execution** because known early at compile time.
- 6) Compile time polymorphism is **less flexible** as all things execute at compile time.

### Run time Polymorphism

- 1) In Run time Polymorphism, call is **not** resolved by the compiler.
- 2) It is also known as **Dynamic binding**, **Late binding** and **overriding** as well.
- 3) **Overriding** is run time polymorphism having same method with same parameters or signature, but associated in a class & its subclass.
- 4) It is achieved by **virtual functions** and **pointers**.
- 5) It provides **slow execution** as compare to early binding because it is known at runtime.
- 6) Run time polymorphism is **more flexible** as all things execute at run time.

Source : <http://freefeast.info/difference-between/difference-between-runtime-polymorphism-and-compile-time-polymorphism/>

**Q 9 :** Describe different ways of Method Overloading in Java .

**Ans :**

Method 1: By changing the number of parameters.

Method 2: By changing the Data types of the parameters

Method 3: By changing the Order of the parameters

Source : <https://www.geeksforgeeks.org/different-ways-method-overloading-java/>

**Q 10 :** Define the use of Super keyword in overriding .

**Ans :** The overridden method in the subclass is executed by default. It is a common practice to invoke an overridden method in the superclass from within the subclass. We can invoke an overridden method in the superclass from within the subclass by using the *super* keyword followed by the dot operator and then the method name.

Source : <http://www.javacjava.com/SuperOverride.html>

**Q 11:**How dynamic polymorphism solve memory problem of static polymorphism in java ?

**Ans :**Static polymorphism:

The process of binding the overloaded method within object at compile time is known as Static polymorphism due to static polymorphism utilization of resources (main memory space) is poor because for each and every overloaded method a memory space is created at compile time when it binds with an object.

Dynamic polymorphism :

In dynamic polymorphism method of the program binds with an object at runtime the advantage of dynamic polymorphism is allocating the memory space for the method (either for overloaded method or for override method) at run time.

Source : <https://www.sitesbay.com/java/java-polymorphism>