|  |  |
| --- | --- |
| **Method overloading** | **Method overriding** |
| 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. |
| Method overloading is performed within class. | Method overriding occurs in two classes that have IS-A (inheritance) relationship. |
| In case of method overloading, parameter must be different. | In case of method overriding, parameter must be same. |
| Method overloading is the example of compile time polymorphism. | Method overriding is the example of run time polymorphism. |
| 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 overriding. |
| Method overloading is a compile-time polymorphism. | Method overriding is a run-time polymorphism. |
| Method overloading may or may not require inheritance. | Method overriding always needs inheritance. |
| Static binding is being used for overloaded methods. | Dynamic binding is being used for overriding methods. |
| Poor Performance due to compile time polymorphism. | It gives better performance. The reason behind this is that the binding of overridden methods is being done at runtime. |
| Private and final methods can be overloaded. | Private and final methods can’t be overridden. |