

1. Overload – Same method name but different parameters, occurs in methods of the same class.

Overriding - When the method signature (name and parameters) are the same in the superclass and the child class, it's called Overriding, occurs between superclass and subclass.

With Overloading, the method to call is determined at the compile-time. With overriding, the method call is determined at the runtime based on the object type.

If overriding breaks, it can cause serious issues in our program because the effect will be visible at runtime. Whereas if overloading breaks, the compile-time error will come and it's easy to fix.

2. A **static method** is a method that belongs to a class, but it does not belong to an instance of that class and this method can be called without the instance or object of that class.

Non-static methods can access any **static** method and **static** variable, without creating an instance of the object.

In **non-static** method, the method can access static data members and static methods as well as non-static members and method of another class or same class, also can change the values of any static data member.

3. Java Constructor overloading is a technique in which a class can have any number of constructors that differ in parameter list. The compiler differentiates these constructors by taking into account the number of parameters in the list and their type.

4. A final class cannot be instantiated, a final method cannot be overridden and a final variable cannot be reassigned.

5. The **finally** keyword is used to create a block of code that follows a try block. A finally block of code always executes, whether or not an exception has occurred. Using a finally block allows you to run any cleanup-type statements that you just wish to execute, despite what happens within the protected code.

6.

Abstract class	Interface
1) Abstract class can have abstract and non-abstract methods.	Interface can have only abstract methods. Since Java 8, it can have default and static methods also.
2) Abstract class doesn't support multiple inheritance .	Interface supports multiple inheritance .
3) Abstract class can have final, non-final, static and non-static variables .	Interface has only static and final variables .
4) Abstract class can provide the implementation of interface .	Interface can't provide the implementation of abstract class .
5) The abstract keyword is used to declare abstract class.	The interface keyword is used to declare interface.
6) An abstract class can extend another Java class and implement multiple Java interfaces.	An interface can extend another Java interface only.
7) An abstract class can be extended using keyword "extends".	An interface can be implemented using keyword "implements".
8) A Java abstract class can have class members like private, protected, etc.	Members of a Java interface are public by default.
9) Example: <pre>public abstract class Shape{ public abstract void draw(); }</pre>	Example: <pre>public interface Drawable{ void draw(); }</pre>

Simply, abstract class achieves partial abstraction (0 to 100%) whereas interface achieves fully abstraction (100%).

7. Git is software for tracking changes in any set of files, usually used for coordinating work among programmers collaboratively developing source code during software development. Its goals include speed, data integrity, and support for distributed, non-linear workflows (thousands of parallel branches running on different systems).

Git (the main implementation in C) is primarily developed on Linux, although it also supports most major operating systems, including BSD, Solaris, macOS, and Windows.

8. A Git repository is the . git/ folder inside a project. This repository tracks all changes made to files in your project, building a history over time.

9. Maven is a tool that can be used for building and managing any Java-based project. Maven make the day-to-day work of Java developers easier and generally help with the comprehension of any Java-based project.

1. We can easily build a project using maven.

2. We can add jars and other dependencies of the project easily using the help of maven.
3. Maven provides project information (log document, dependency list, unit test reports etc.)
4. Maven is very helpful for a project while updating central repository of JARs and other dependencies.
5. With the help of Maven we can build any number of projects into output types like the JAR, WAR etc without doing any scripting.
6. Using maven we can easily integrate our project with source control system (such as Subversion or Git).

10. API is the acronym for Application Programming Interface, which is a software intermediary that allows two applications to talk to each other. Each time you use an app like Facebook, send an instant message, or check the weather on your phone, you're using an API.

In building applications, an API (application programming interface) simplifies programming by abstracting the underlying implementation and only exposing objects or actions the developer needs.