Super Keyword

The **super** keyword in Java is a reference variable which is used to refer immediate parent class object.

Whenever you create the instance of subclass, an instance of parent class is created implicitly which is referred by super reference variable.

Usage of Java super Keyword

1. super can be used to refer immediate parent class instance variable.
2. super can be used to invoke immediate parent class method.
3. super() can be used to invoke immediate parent class constructor.

this keyword

this is a reference variable that refers to the current object on which the method or constructor is being invoked

Usage of Java super Keyword

1. Using this() to invoke the current class constructor
2. this is used to refer current class instance variable

final keyword

The **final keyword** in java is used to restrict the user. The java final keyword can be used in many context. Final can be:

1. variable (Constant veriable)
2. method (you cannot override it.)
3. class (Can't be Inherite)

Java static keyword

The static keyword in Java is used for memory management mainly. We can apply static keyword with variables, methods, blocks and nested classes.

The static can be:

1. Variable (also known as a class variable)
2. Method (also known as a class method)
3. Block

Static keyword in Java - Javatpoint

Inheritance

Inheritance in Java is a mechanism in which one object acquires all the properties and behaviors of a parent object. It is an important part of OOPs (Object Oriented programming system).

The **extends keyword** indicates that you are making a new class that derives from an existing class. The meaning of "extends" is to increase the functionality.

Types of inheritance in java

- **Single Inheritance**
- **Multilevel Inheritance**
- **Hierarchical Inheritance**

Why multiple inheritance is not supported in java?

→ To reduce the complexity and simplify the language, multiple inheritance is not supported in java.

Consider a scenario where A, B, and C are three classes. The C class inherits A and B classes. If A and B classes have the same method and you call it from child class object, there will be ambiguity to call the method of A or B class.

Polymorphism in Java

Polymorphism in Java is a concept by which we can perform a *single action in different ways*. Polymorphism is derived from 2 Greek words: poly and morphs. The word "poly" means many and "morphs" means forms. So polymorphism means many forms. We can perform polymorphism in java by method overloading and method overriding.

Types of polymorphism

1. Runtime Polymorphismt
2. compile-time polymorphism

Features of Java

- Simple
- Object Oriented
- MultiTasking
- Portable
- Platform Independent
- Secured
- High Performance
- Dynamic
- Support Documentation

In Java, **literals** are the constant values that appear directly in the program. It can be assigned directly to a variable. Java has various types of literals. The following figure represents a literal

Eg. Int a=500