**Polymorphism** in java – is viewed as an essential element of OOP. Polymorphism is the ability to perform a single action in multiple ways.

Polymorphism is derived from Greek word: POLY means – MANY & MORPH means – FORMS (many forms).

EX: - a person may have multiple characteristics at the same time.

A woman for ex: - is a mother, sister, daughter, worker and a wife all at the same time.

Types of polymorphism

1. **Compile time polymorphism** – is also known as static polymorphism, is a type of polymorphism that occurs during the compilation. We can use method overloading to accomplish it.

Method overloading – when several methods with the same name but different parameters exists in a class, these methods are said to be overloaded. The key benefit is that it improves program readability. Methods can be overloaded by using different numbers of arguments or (and) by using different types of arguments.

* 1. **With different number of arguments** – isme parameters k arguments ko badal kar kaam kiya ja sakta h
     1. Jaise:-
        1. multiply(int a, int b)
        2. Multiply(int a, int b, int c)
  2. **With different type of arguments** – isme argumemts k data type ko badal ka kaam kiya ja sakta h
     1. Jaise:-
        1. multiply(int a, int b)
        2. multiply(double a, double b)

1. **Runtime polymorphism** – dynamic polymorphism is another name of runtime polymorphism. In java, method overriding is a technique for implementing runtime polymorphism. It is also sometimes called dynamic method dispatch.

**Method overriding** – is a feature that allows you to redefine a base class method in the derived class depending on the requirement of the derived class. In other words, whatever methods are available in the base class are by default also available in the derived class. However, a derived class can dissatisfied with the base class method implementation at times. Then, the derived class can redefine the method based on its requirements.

**Rules of method overriding** –

* One class (derived class) should inherit another class (base class)
* The base class’s method and the derived class’s method must have the same name.
* The parameters of the base class’s method and the deriver class’s method must be same.

**Method hiding** in java – if a subclass defines a static method with the same signature as a static method in a superclass, the subclass method hides the superclass method. It is only possible to hide methods if both the parent and child classes have static method.