Method overloading in Java allows a class to have multiple methods with the same name, provided their parameter lists are different. meaning the compiler determines which method to call based on the method signature at compile time.

Key Points of Method Overloading:

* **Same Method Name:**Overloaded methods must share the same name within the same class.
* **Different Parameter Lists:**The crucial distinction between overloaded methods lies in their parameter lists. This difference can be achieved by:
  + **Different Number of Parameters:** For example, add(int a, int b) and add(int a, int b, int c).
  + **Different Types of Parameters:** For example, add(int a, int b) and add(double a, double b).
  + **Different Order of Parameters:** For example, display(int a, String s) and display(String s, int a).
* **Return Type Flexibility:**Overloaded methods can have the same or different return types. However, only changing the return type without altering the parameter list is not considered method overloading. The parameter list must differ for overloading to occur.
* **Access Modifier Flexibility:**Overloaded methods can have different access modifiers (e.g., public, private, protected).
* **Exception Handling:**Overloaded methods can declare different checked exceptions or no exceptions at all.

Advantages:

* **Improved Readability and Reusability:**Allows for methods performing similar operations to share a common name, making the code easier to understand.
* **Reduced Complexity:**Simplifies the API design by allowing a single method name to handle various input scenarios.
* **Enhanced Flexibility:**Provides different ways to perform an operation based on the input types or number of arguments.