- Method overloading within objects, a key aspect of polymorphism in Object-Oriented Programming (OOP), allows a class to define multiple methods with the same name but different parameter lists. This enables a single method name to perform different actions based on the types or number of arguments provided during the method call.
- When an overloaded method is called, the compiler determines which specific version to execute based on the arguments passed during the call. It matches the provided arguments to the parameter lists of the overloaded methods to find the most appropriate one.

Example by Method Overloading in PrintStream:

Method Signature	Argument Type	Purpose
println(String x)	String	Prints the string directly.
println(int x)	int	Prints the integer value.
println(double x)	double	Prints the double value.
println(boolean x)	boolean	Prints "true" or "false".
println(Object x)	Any non-primitive type (like List, ArrayList, or any custom class)	This is the catch-all for objects. It relies on the object's toString() method.