1.How to create an object in Java?
Ans To create an object in Java, you need to follow these steps:

1. Define a class: A class is a blueprint that defines the properties and behaviours of an object. You can define a class using the `class` keyword.
2. Declare an object reference variable: To create an object, you need to declare a variable of the class type. This variable will hold the reference to the object.
3. Instantiate the object: Use the `new` keyword followed by the class name and parentheses to create an instance of the class
4. Access the object's members: Once the object is created, you can access its members (fields and methods) using the dot operator (`‘-’).`

2.What is the use of the new keyword in Java?
Ans In Java, the `new` keyword is used to create objects. It's like a magic word that brings objects to life! When you want to use a class to create something tangible in your program, you use `new`.

Here's a simple way to understand its use:

1. Imagine you have a blueprint (class) for a car, with all its properties and behaviours.
2. To actually create a real car (object) based on that blueprint, you use the `new` keyword followed by the class name and parentheses.
3. Once you have the car object, you can access its features, modify its properties, and make it do things by using dot notation.

   In summary, the `new` keyword is used to create objects from class blueprints. It's how you bring your classes to life and work with them in your program, setting their properties and making them perform actions.

3.What are the different types of the variables in java?
Ans. In Java, there are three main types of variables: local variables, instance variables, and static variables.

4.What is the difference between Instance variable and local variable?
Ans.

Instance Variable:

- Belongs to an object of a class.
- Can be accessed and modified by any method within the class.
- Exists as long as the object exists.

- Used to store data that is shared among different methods of the class.

Local Variable:

- Belongs to a specific method or block.
- Can only be accessed within that method or block.
- Exists only as long as the method or block is executing.
- Used for temporary storage or intermediate calculations within a specific scope.

5.In which area memory is allocated for instance variable and local variable?
Ans.

- Instance variables are stored in memory on the heap and exist as long as the object exists.
- Local variables are stored on the stack and exist only within the scope of the method or block where they are declared.

6.What is method overloading?
Ans. the ability to have multiple methods with the same name but different parameters within a class. It allows you to define multiple versions of a method, each accepting different input parameters or having a different return type.

Here's an easy-to-understand explanation of method overloading:

Imagine you have a class called "Calculator" and you want to provide different ways to add numbers. With method overloading, you can define multiple methods named "add" but with different parameter lists.