29th May Java Objects and Methods

1. **How to Create an Object in Java?**

Ans: To create an object in Java, you need to follow these steps:

a. Define a class: Start by defining a class that describes the properties and behaviour of the object you want to create.

b. Instantiate the object: Use the "new" keyword followed by the class name and parentheses to create a new instance of the class.

c. Initialize the object: Once the object is created, you can use the dot notation to access its methods and set its properties.

Here's an example of creating an object of class "Person":

```
public class Person {
  String name;
  int age;

  public void speak() {
    System.out.println("Hello, my name is " + name + " and I'm " + age + " years old.");
  }
}

public class Main {
  public static void main(String[] args) {
    Person person = new Person();
    person.name = "John";
    person.age = 25;
    person.speak();
  }
```

}

2. **What is the use of a new keyword in Java?**

Ans: The "new" keyword in Java is used to create an instance (object) of a class. When you use the "new" keyword, memory is allocated to store the object, and the constructor of the class is called to initialize the object. It is an essential part of the process of creating objects in Java.

3. **What are the different types of variables in Java?**

Ans: In Java, there are three types of variables:

a. Local variables: These variables are declared within a method, constructor, or block and have a limited scope. They are only accessible within the block where they are declared.

b. Instance variables: These variables are declared within a class but outside any method, constructor, or block. Each instance of the class (object) has its own copy of instance variables.

c. Class variables (static variables): These variables are declared with the "static" keyword within a class but outside any method, constructor, or block. They belong to the class itself rather than to any specific instance of the class, and all objects of the class share the same copy of the variable.

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

Ans: The main differences between instance variables and local variables are as follows:

Scope: Instance variables are accessible throughout the entire class, while local variables are only accessible within the method, constructor, or block where they are declared.

Lifetime: Instance variables exist as long as the object to which they belong exists. Local variables have a shorter lifetime and are created and destroyed each time the method, constructor, or block is executed.

Initialization: Instance variables are initialized with default values if not explicitly assigned, while local variables must be explicitly initialized before they can be used.

Memory Allocation: Instance variables are stored in memory along with the object they belong to, whereas local variables are stored on the stack.

5. **In Which area memory is allocated for instance variable and local variable?**

Ans: Memory for instance variables is allocated in the heap memory when an object is created using the "new" keyword. Each instance variable has its own memory space allocated within the object.

Memory for local variables is allocated on the stack when the method, constructor, or block is executed. It is released when the method, constructor, or block execution completes.

6. **What is method overloading?**

Ans: Method overloading in Java refers to the ability to define multiple methods with the same name but with different parameters within a class. The methods must have different

parameter lists (different number of parameters or different types of parameters). Method overloading allows you to provide different ways to perform similar operations based on the type or number of arguments passed to the method.