

1. How to Create an Object in Java?

Ans: First, a class must be created for the JVM to create an object. Then, add the main method inside the class. Some attributes and behaviors of the object are defined. Finally, the object is created using two steps, first is to declare the variable using the class name and a reference variable, and then to create the object using the new keyword and the name of the class as a method.

Example: `Test object = new Test();`

// In the above line, Test is the name of the class and object is the reference variable

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

Ans: new keyword helps in creating an object of a specific class by reserving an address block in the Heap area where the values of the object are stored. When the object is created, default values are stored in the address block.

It is used in code in the following manner:

Eg: `Test2 object1 = new Test2();`

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

Ans: There are 3 different types of variables in Java they are Local variables, Instance variables, and Static variables.

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

Ans: Local variables are declared inside a method/constructor/block and their scope is dependent on the method's/constructor's/block's scope, i.e, they are created(allocated a space in Stack Area) when the method/constructor/block is called and destroyed(removed the allocated space from Stack Area) when the method/constructor/block call returns. It is mandatory to initialize a local variable before using it.

Instance variables are declared outside a method/constructor/block but inside the class, i.e, they are created when the object is created(object values[instance variables] are stored in the Heap area), and destroyed(object values[instance variables] are removed from Heap Area and collected by Garbage collector) when the object is destroyed. Their initialization is not mandatory. However, instance variables can not be accessed directly, they have to be accessed using the specific object.

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

Ans: For local variables, some space in the Stack area is allocated. For instance variables, some space in the Heap area is allocated.

6. What is method overloading?

Ans: Method overloading is the process of using the same method names but with distinct signatures. When we say signatures, it means that they might vary based on the quantity(one or many), nature(int/float, etc.), or combination of input arguments.

Example:

`add()` is a method to add numbers.

```
// Method 1
public int add(int n1, int n2){
    int n = n1 + n2;
    return n;
}
```

```
// Method 2
public int add(int n1, int n2, int n3){
    int n = n1 + n2 + n3;
    return n;
}
```

Even though methods 1 and 2 have the same name, the Compiler won't give any error as the signature of the two methods is different, i.e, Compiler understands Method Overloading concept. If methods are written following Method Overloading wherever necessary, it reduces the Complexity of Programming and makes documentation easier.