Oops Fundamentals

1. **How to create an object in Java?**

   Answer: Java provides five ways to create an object.

   - Using new Keyword.
   - Using clone() method
   - Using newInstance() method of the Class class.
   - Using newInstance() method of the Constructor class.
   - Using Deserialization

   Generally, we use '**new**' keyword for creating an object in Java.

   Using the **new** keyword is the most popular way to create an object or instance of the class. When we create an instance of the class by using the new keyword, it allocates memory (heap) for the newly created **object** and also returns the **reference** of that object to that memory. The new keyword is also used to create an array.

   Ex.- **className object = new className();**

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

   Answer: The Java new keyword is used to create an instance of the class. In other words, it instantiates a class by allocating memory for a new object and returning a reference to that memory. We can also use the new keyword to create the array object.

   - It is used to create the object.
   - It allocates the memory at runtime.
   - All objects occupy memory in the heap area.
   - It invokes the object constructor.
   - It requires a single, postfix argument to call the constructor

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

   Answer: There are three types of variables in java:

   - Local variable
   - Instance variable
   - Static variable

- **Local Variable:**

  A variable declared inside the body of the method is called local variable. You can use this variable only within that method and the other methods in the class aren't even aware that the variable exists.

  A local variable cannot be defined with "static" keyword.

- **Instance Variable**

  A variable declared inside the class but outside the body of the method, is called an instance variable. It is not declared as static.

  It is called an instance variable because its value is instance-specific and is not shared among instances.

- **Static variable:**

  A variable that is declared as static is called a static variable. It cannot be local. You can create a single copy of the static variable and share it among all the instances of the class. Memory allocation for static variables happens only once when the class is loaded in the memory.

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

   Answer:

   Instance Variable: These variables are declared within a class but outside a method, constructor, or block and always get a default value.

   - These variables are usually created when we create an object and are destroyed when the object is destroyed.

   - We may use an access specifier, for instance, variable, and if no access specifier is specified, then the default access specifier is used.

   - Each and every object will have its own copy of instance variables.
   - These variables are destroyed when the object is destroyed.
   - It can be accessed throughout the class.
   - It is not compulsory to initialize instance variables before use.

- It includes access modifiers such as private, public, protected, etc.

Local Variable: These variables are declared within a method but do not get any default value.

- They are usually created when we enter a method or constructor and are destroyed after exiting the block or when the call returns from the method.

- Its scope is generally limited to a method and its scope starts from the line they are declared. Their scope usually remains there until the closing curly brace of the method comes.

- The initialization of the local variable is mandatory.
- These variables are destroyed when the constructor or method is exited.
- Its access is limited to the method in which it is declared.
- It is important to initialize local variables before use.

- It does not include any access modifiers such as private, public, protected, etc.

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

   Answer: For the instance variable memory is allocated in the heap area. And for the local variable memory is allocated in the stack area.

6. **What is method overloading?**

   Answer: If a class has multiple methods having same name but different in parameters, it is known as Method Overloading.

   If we have to perform only one operation, having same name of the methods increases the readability of the program.

Suppose you have to perform addition of the given numbers but there can be any number of arguments, if you write the method such as a(int,int) for two parameters, and b(int,int,int) for three parameters then it may be difficult for you as well as other programmers to understand the behavior of the method because its name differs.

So, we perform method overloading to figure out the program quickly.

Advantage of method overloading

Method overloading *increases the readability of the program.*
Different ways to overload the method.

There are two ways to overload the method in java

1. By changing number of arguments
2. By changing the data type