1.How to create an object in java?

Ans. The **object** is a basic building block of an [OOPs](https://www.javatpoint.com/java-oops-concepts) language. In **Java**, we cannot execute any program without creating an **object.** There is various way to **create an**[object in Java](https://www.javatpoint.com/object-and-class-in-java) that we will discuss in this section, and also learn **how to create an object in Java.**

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

2.What is the use of new keyword in java?

Ans. Here are the some of uses of new keyword in Java:

* **Memory Allocation:** The new keyword is used to allocate memory for an object on the heap, the memory space where objects are stored in Java. Without the "new" keyword, an object cannot be created and there would be no memory allocated for it.
* **Object Initialization:** The new keyword in java also calls the constructor of a class to initialize the object’s state. The constructor sets the initial values for the object’s fields and performs any other necessary setup. Without the "new" keyword, the constructor cannot be called and the object would not be properly initialized.
* **Object Polymorphism:** The new keyword in java is used to create instances of subclasses that inherit from a superclass, the runtime type of the object will be the type of the subclass, making the object able to use the method overridden by the subclass and not the one inherited from the superclass.
* **Dynamic Memory Allocation:** the new Keyword in Java can be used to create arrays dynamically, which means the size of the array can be determined during runtime.

3.What are the different type of variables in java?

* Ans. Local variable: These variables are declared and used within a specific method, constructor, or block of code. They are only accessible within the scope in which they are defined. Local variables must be initialized before they can be used.
* Instance variable: These variables are declared within a class but outside any method, constructor, or block of code. They are associated with individual instances (objects) of the class. Each instance of the class has its own copy of the instance variables. Instance variables are initialized to their default values if not explicitly assigned.
* Static variable: These variables are declared with the static keyword within a class but outside any method, constructor, or block of code. They are associated with the class itself rather than with instances of the class. Static variables are shared among all instances of the class, and changes to the variable will be reflected in all instances. Static variables are initialized to their default values if not explicitly assigned.

4.What is the difference between instance variable and local varible?

Ans.

|  |  |
| --- | --- |
| Local variable | Instance Variable |
| 1. Variables declared within a method are local variables. | 1. An instance variable is declared inside a class but outside of any method or block. |
| 2. The scope of the local variable is limited to the method it is declared inside. | 2. An instance variable is accessible throughout the class. |
| 3. A local variable starts its lifetime when the method is invoked. | 3. The object associated with the instance variable decides its lifetime. |
| 4. Local variable is accessible to all the objects of the class. | 4. Instance variable has different copies for different objects. |
| 5. Used to store values that are required for a particular method. | 5. Used to store values that are needed to be accessed by different methods of the class |

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

Ans**. Stack** is a memory place where the methods and the local variables are stored. (variable references either **primitive**or**object** references are also stored in the stack)**. Heap** is a memory place where the objects and its instance variable are stored.

6.What is method overloading?

Ans. If a [class](https://www.javatpoint.com/object-and-class-in-java) 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](https://www.javatpoint.com/java-programs). 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.