Oops Fundamentals

1.How to create an object in Java?

The object is a basic building block of an OOPs language. In Java, we cannot execute any program without creating an object.

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

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

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.

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

A variable is a container which holds the value while the Java program is executed. A variable is assigned with a data type.

Variable is a name of memory location. There are three types of variables in java: local, instance and static.

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

Instance variables are declared within classes outside methods, and they store the state of an object, while local variables are declared within code blocks, and they are used for storing the state of a method. An instance variable is live as long as the object that contains that variable is live, while a local variable is live during the execution of that method/ code block. An instance variable (that is declared public) can be accessed within the class, whereas a local variable can only be accessed within the code block that it is declared. Usage of instance variables is only limited to object oriented programming, while local variables do not have such a limitation.

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

Stack is a memory place where the methods and the local variables are stored.

Heap is a memory place where the objects and its instance variable are stored.

6.What is Method overloading?

Method overloading in java is a feature that allows a class to have more than one method with the same name, but with different parameters.

Java supports method overloading through two mechanisms:

1. By changing the number of parameters
2. By changing the data type of parameters Overloading by changing the number of parameters A method can be overloaded by changing the number of parameters.