Instantiation
In Java, instantiation means to call the constructor of a class that creates an instance or object of that class type. In other words, creating an object of the class is called instantiation. It occupies the initial memory for the object and returns a reference. An object instantiation in Java provides the blueprint for the class.

Syntax for Instantiation:
ClassName objName = new ClassName();
Or
ClassName cn;
cn= new ClassName;

There are two ways to create instances:
->Using the new Keyword
->Using Static Factory Method

Heap and Stack Memory -
Stack Memory:
The stack memory is a physical space (in RAM) allocated to each thread at run time. It is created when a thread creates. Memory management in the stack follows LIFO (Last-In-First-Out) order because it is accessible globally. It stores the variables, references to objects, and partial results. Memory allocated to stack lives until the function returns.

Heap Memory:
It is created when the JVM starts up and is used by the application as long as it runs. It stores objects and JRE classes. Whenever we create objects it occupies space in the heap memory while the reference of that object creates in the stack. It does not follow any order like the stack. It dynamically handles the memory blocks.

Garbage Collection :
Garbage collection in Java is the process by which Java programs perform automatic memory management. Java programs compile to bytecode that can be run on a Java Virtual Machine, or JVM for short. When Java programs run on the JVM, objects are created on the heap, which is a portion of memory dedicated to the program. Eventually, some objects will no longer be needed. The garbage collector finds these unused objects and deletes them to free up memory.