ASSIGNMENT

1.what is garbage collector and how it works?

ANS:Garbage Collection is the process of reclaiming the runtime unused memory by destroying the unused objects.

Java Garbage Collection is the process by which Java programs perform automatic memory management. Java programs compile into bytecode that can be run on a Java Virtual Machine (JVM).

When Java programs run on the JVM, objects are created on the heap, which is a portion of memory dedicated to the program.

Over the lifetime of a Java application, new objects are created and released. Eventually, some objects are no longer needed. You can say that at any point in time, the heap memory consists of two types of objects:

* *Live* - these objects are being used and referenced from somewhere else
* *Dead* - these objects are no longer used or referenced from anywhere

The garbage collector finds these unused objects and deletes them to free up memory.

Java garbage collection is an automatic process. The programmer does not need to explicitly mark objects to be deleted.

The garbage collection implementation lives in the JVM. Each JVM can implement its own version of garbage collection. However, it should meet the standard JVM specification of working with the objects present in the heap memory, marking or identifying the unreachable objects, and destroying them with compaction.

2.what is heap space?

**ANS: The Java heap is the area of memory used to store objects instantiated by applications running on the JVM.**When the JVM is started, heap memory is created and any objects in the heap can be shared between threads as long as the application is running. The size of the heap can vary, so many users restrict the Java heap size to 2-8 GB in order to minimize garbage collection pauses.

3. what is metaspace?

ANS: Meta space is a new memory space – starting from the Java 8 version; it has replaced the older PermGen memory space. The most significant difference is how it handles memory allocation. Specifically, this native memory region grows automatically by default. The OpenJDK uses Meta space to store its class metadata. It can contribute a large part to the non-Java-heap memory footprint of a Java VM process. Meta space is memory the VM uses to store class Metadata. Metadata is defined as data about the data. It is the documentation about the information which is required by the users. It is used to analyze data usage and performance of particular data.

4. what is java memory model?

ANS: The Java memory model specifies how the Java virtual machine works with the computer's memory (RAM). The Java virtual machine is a model of a whole computer so this model naturally includes a memory model.

Java memory model is divided between Thread Stacks (One for each thread) and a heap area.

Thread Stack: It is a thread specific memory area and contains local variables, methods call information etc. JVM stacks could be of fixed size or variable size. If computation in a thread exceeds its stack size limit then JVM throws StackOverflowError and exits.

**Heap**

It contains all the objects created during application lifecycle. The heap is created when the virtual machine starts up. Garbage collector reclaims heap storage for objects and objects are never explicitly deallocated. The JVM is not using any automatic storage management system, and it may vary as per the system requirements. The heap may be of a fixed size or may vary as per requirement. The memory for the heap does not need to be contiguous.

5. what is young and old generations?

ANS: The Young Generation

From a high level, the **young generation** is where all new objects start out. Once they’re allocated in the Java code, they go specifically to this subsection called the **eden space**.

Eventually, the eden space fills up with objects. At this point, a **minor garbage collection**event occurs.

That’s where the **marking algorithm** I described earlier comes into play. Some objects (those that are **referenced**) are marked, and some (those that are **unreferenced**) are not. Those that had been marked then move onto another subsection of the young generation called **S0** of the **survivor space**(note that the survivor space itself is split into two parts, S0 and S1). Those left unmarked are cleared out by Java’s automatic garbage collection.

The Old Generation

The old generation can be thought of as **where long-lived objects lie**. Basically, if objects **reach a certain age threshold** **after multiple garbage collection events in the young generation**, then they can then be moved to the old generation.

When objects get garbage collected from the old generation, a **major garbage collection event** occurs. The old generation is comprised of only one section called the **tenured generation**. This is why in conversation or in reading sometimes that the two terms have come to be mostly interchangeable.

6. what is eden and survivor space ?

ANS: Eden Space, Survivor Space, Tenured Space are part of heap memory Metaspace and Code Cache are part of non-heap memory. Codecache: The Java Virtual Machine (JVM) generates native code and stores it in a memory area called the codecache.