What are the advantages of Java in software development?
One of the advantages of developing an java application is multi-platform. That is you can write an application that runs on any operating system in which there is a Java Runtime Environment.

What is stack memory and heap? Which one is used for what?
A Java **stack** is part of your computer's memory where variables, which are created by all functions you do, are stored. It is used to execute a thread and may have certain short-lived values as well as references to other objects.

Java objects are in an area, which is called the **heap**. Java Heap space is used by java runtime to allocate memory to Objects and JRE classes. Whenever we create any object, it's always created in the Heap space. It is created when the program is run, and its size may decrease or increase as your program runs. This is when objects that are no longer used are deleted to make way for new objects.

Based on the above explanations, we can **conclude differences between** Heap and Stack memory.

1. Heap memory is used by all the parts of the application whereas stack memory is used only by one thread of execution.
2. Whenever an object is created, it's always stored in the Heap space and stack memory contains the reference to it. Stack memory only contains local primitive variables and reference variables to objects in heap space.
3. Objects stored in the heap are globally accessible whereas stack memory can't be accessed by other threads.
4. Stack memory is short-lived whereas heap memory lives from the start till the end of application execution.

Please explain parameters: -Xmx / -Xms / -Xss / -Xmn

- **-Xmx** specifies the maximum memory size, above which the application does not use
- **-Xms** determines the size of the initial allocated memory for objects
- **-Xss** specifies the largest stack size of a particular thread.
- **-Xmn** specifies the amount of memory used up to which the garbage collector should free up memory (if possible);

What is GC process? What does it needed for?
Java **garbage collection** 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. When Java programs run on the JVM, objects are created on the heap, which is a portion of memory. In the end, some objects will no longer be needed. The garbage collector finds these unused objects and deletes them to free up memory.

What is New Generation (Eden, 2 Survivors) and Old Generation?

- **Young Generation**, It is place where lived for short period and divided into two spaces:
- **Old Generation**, This pool contain limited and virtual (reserved) space and will be holding those objects which survived after garbage collection from Young Generation.

Please describe algorithms used by GCs you know?

In simplest case, Basic process of Garbage collection can be described as follows:

1. Marking. Marking means identifies which pieces of memory are in use and which are not.
2. Normal Deletion. In this phase, normal deletion removed the unreferenced objects leaving referenced object and pointers to free space
3. Deletion with compacting. in addition to deleting unreferenced objects, you can compact the remaining referenced objects. By doing this, this makes memory allocation much easier and faster.

Why does Java use different algorithms for old and young generation objects?

it uses to enhance the performance of the JVM. Therefore, the heap is broken up into smaller parts or generations. The heap parts are: young, old generation.