Answer these questions about the jvm -

1.What is the difference between the stack and heap memory areas allocated in JVM?

Heap memory is used by all parts of the jvm, while stack memory is only used by one thred.

2.Can you explain the architecture of JVM?

JVM contains a classLoader, a memory area, execution engine etc.

Classloader is a subsystem of JVM which is used to load class files. When we run a java program, its loaded first in classLoader.  
Class(Method) Area stores per-class structures such as the runtime constant pool, field and method data, the code for methods.  
Heap

It is the runtime data area in which objects are allocated.

Stack

Java Stack stores frames. It holds local variables and partial results, and plays a part in method invocation and return.  
Each thread has a private JVM stack, created at the same time as thread.

A new frame is created each time a method is invoked. A frame is destroyed when its method invocation completes.

Program Counter Register

PC (program counter) register contains the address of the Java virtual machine instruction currently being executed.

Native Method Stack

It contains all the native methods used in the application.



3.Why not just run native? why is the jvm needed?   
We need to use the JVM in order to enable our computer to run and execute java bytecode.

4.What do you understand about bytecode in Java?

5.Can you tell me more about the memory areas available in JVM?

6.What is just-in-time compilation?