JAVA =>
1) What is the platform?
->A platform is the hardware or software environment in which a piece of software is executed. There are two types of platforms, software-based and hardware-based. Java provides the software-based platform.

2) What are the main differences between the Java platform and other platforms?
->There are the following differences between the Java platform and other platforms.
* Java is the software-based platform whereas other platforms may be the hardware platforms or software-based platforms.
* Java is executed on the top of other hardware platforms whereas other platforms can only have the hardware components.

3) What gives Java its 'write once and run anywhere' nature?
->The bytecode. Java compiler converts the Java programs into the class file (Byte Code) which is the intermediate language between source code and machine code. This bytecode is not platform specific and can be executed on any computer.

4) What is classloader?
->Classloader is a subsystem of JVM which is used to load class files. Whenever we run the java program, it is loaded first by the classloader. There are three built-in classloaders in Java.
1. Bootstrap ClassLoader: This is the first classloader which is the superclass of Extension classloader. It loads the rt.jar file which contains all class files of Java Standard Edition like java.lang package classes, java.net package classes, java.util package classes, java.io package classes, java.sql package classes, etc.
2. Extension ClassLoader: This is the child classloader of Bootstrap and parent classloader of System classloader. It loads the jar files located inside $JAVA_HOME/jre/lib/ext directory.
3. System/Application ClassLoader: This is the child classloader of Extension classloader. It loads the class files from the classpath. By default, the classpath is set to the current directory. You can change the classpath using "-cp" or "-classpath" switch. It is also known as Application classloader.

5) Is delete, next, main, exit or null keyword in java?
->No.

6) If I don't provide any arguments on the command line, then what will the value stored in the String array passed into the main() method, empty or NULL?
->It is empty, but not null.

7) What if I write static public void instead of public static void?
->The program compiles and runs correctly because the order of specifiers doesn't matter in Java.

===

OS=>

1) What is the use of paging in operating system?
Paging is used to solve the external fragmentation problem in operating system. This technique ensures that the data you need is available as quickly as possible.

2) What is the concept of demand paging?
Demand paging specifies that if an area of memory is not currently being used, it is swapped to disk to make room for an application's need.

3) What is the advantage of a multiprocessor system?
As many as processors are increased, you will get the considerable increment in throughput. It is cost effective also because they can share resources. So, the overall reliability increases.

4) What is FCFS?
FCFS stands for First Come, First Served. It is a type of scheduling algorithm. In this scheme, if a process requests the CPU first, it is allocated to the CPU first. Its implementation is managed by a FIFO queue.

5) What are the four necessary and sufficient conditions behind the deadlock?
These are the 4 conditions:
i) Mutual Exclusion Condition: It specifies that the resources involved are non-sharable.
ii) Hold and Wait Condition: It specifies that there must be a process that is holding a resource already allocated to it while waiting for additional resource that are currently being held by other processes.
iii) No-Preemptive Condition: Resources cannot be taken away while they are being used by processes.
iv) Circular Wait Condition: It is an explanation of the second condition. It specifies that the processes in the system form a circular list or a chain where each process in the chain is waiting for a resource held by next process in the chain.

===

DS=>

1) How many types of Linked List exist?
There are multiple types of Linked Lists available:

Singly Linked List
Doubly Linked List
Multiply Linked List
Circular Linked List

2) Explain Singly Linked List in short.
The singly linked list includes nodes which contain a data field and next field. The next field further points to the next node in the line of nodes.

In other words, the nodes in singly linked lists contain a pointer to the next node in the list. We can perform operations like insertion, deletion, and traversal in singly linked lists.

A singly linked list is shown below whose nodes contain two fields: an integer value and a pointer value (a link to the next node).

3) What do you understand by Doubly Linked List?
The doubly linked list includes a pointer (link) to the next node as well as to the previous node in the list. The two links between the nodes may be called "forward" and "backward, "or "next" and "prev (previous)." A doubly linked list is shown below whose nodes consist of three fields: an integer value, a link that points to the next node, and a link that points to the previous node.

A technique (known as XOR-linking) is used to allow a doubly-linked list to be implemented with the help of a single link field in each node. However, this technique needs more ability to perform some operations on addresses, and therefore may not be available for some high-level languages.

Most of the modern operating systems use doubly linked lists to maintain references to active threads, processes, and other dynamic objects.

4) Explain Multiply Linked List in short.
In a multiply linked list, each node consists of two or more link fields. Each field is used to join the same set of records in a different order of the same set, e.g. "by name, by date of birth, by the department, etc.".

5) How will you explain Circular Linked List?
In the last node of a linked list, the link field often contains a null reference. Instead of including a null pointer at the end of the list, the last node in circular linked lists includes a pointer pointing to the first node. In such cases, the list is said to be ?circularly linked? otherwise it is said to be 'open' or 'linear.' A circular linked list is that type of list where the last pointer points or contains the address of the first node.

In case of a circular doubly linked list, the first node also points to the last node of the list.

6) Mention a few applications of Linked Lists?
Few of the main applications of Linked Lists are:

Linked Lists let us implement queues, stacks, graphs, etc.
Linked Lists let us insert elements at the beginning and end of the list.

7) How can you insert a node to the beginning of a singly linked list?
==Create a new node
==Insert new node by assigning the head pointer to the new node's next pointer
==Update the head pointer to point the new node

==

==