Q) what is difference between JDK,JRE and JVM?

= The JRE is an abbreviation for Java Runtime Environment. The JVM is an abbreviation for Java Virtual Machine. The JDK (Java Development Kit) is a software development kit that develops applications in Java. Along with JRE, the JDK also consists of various development tools (Java Debugger, JavaDoc, compilers, etc.)

Q) what is JIT compiler ?

= The Just-In-Time (JIT) compiler is a component of the runtime environment that improves the performance of Java™ applications by compiling bytecodes to native machine code at run time.

Q) what is class loader ?

= The **Java ClassLoader** is a part of the **Java Runtime Environment** that dynamically loads Java classes into the **Java Virtual Machine**. The Java run time system does not need to know about files and file systems because of classloaders. Java classes aren't loaded into memory all at once, but when required by an application. At this point, the **Java ClassLoader** is called by the **JRE** and these ClassLoaders load classes into memory dynamically

Q) Explain various memory logical partitions ?

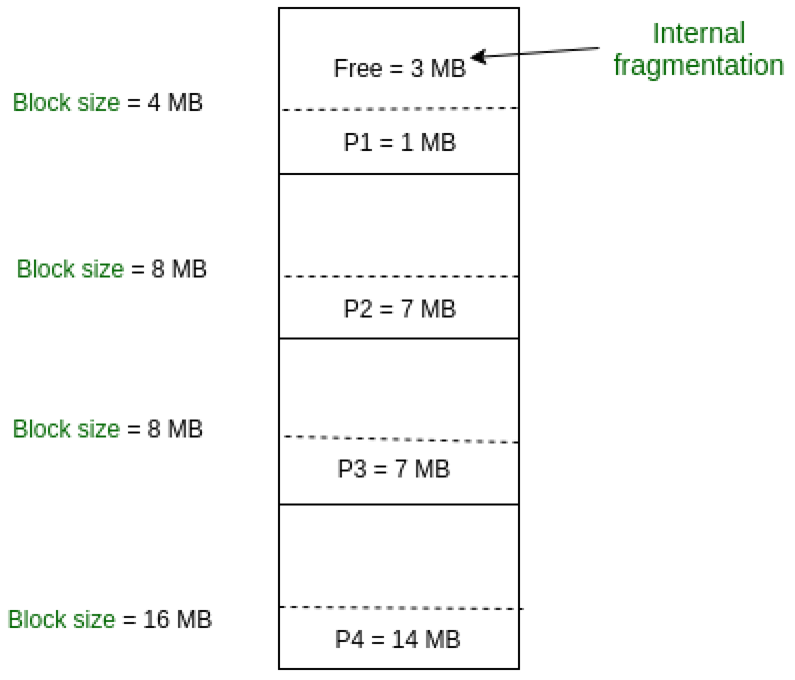
= In operating systems, Memory Management is the function responsible for allocating and managing a computer's main memory. Memory Management function keeps track of the status of each memory location, either allocated or free to ensure effective and efficient use of Primary Memory.

There are two Memory Management Techniques: **Contiguous**, and **Non-Contiguous**. In Contiguous Technique, executing process must be loaded entirely in the main memory. Contiguous Technique can be divided into:

1. Fixed (or static) partitioning

2. Variable (or dynamic) partitioning

Fixed Partitioning:
This is the oldest and simplest technique used to put more than one process in the main memory. In this partitioning, the number of partitions (non-overlapping) in RAM is **fixed but the size** of each partition may or **may not be the same**. As it is a **contiguous** allocation, hence no spanning is allowed. Here partitions are made before execution or during system configure.



Fixed size partition

As illustrated in above figure, first process is only consuming 1MB out of 4MB in the main memory.
Hence, Internal Fragmentation in first block is (4-1) = 3MB.
Sum of Internal Fragmentation in every block = (4-1)+(8-7)+(8-7)+(16-14)= 3+1+1+2 = 7MB.

Suppose process P5 of size 7MB comes. But this process cannot be accommodated in spite of available free space because of contiguous allocation (as spanning is not allowed). Hence, 7MB becomes part of External Fragmentation.

There are some advantages and disadvantages of fixed partitioning.

Advantages of Fixed Partitioning –

1. **Easy to implement:**
   Algorithms needed to implement Fixed Partitioning are easy to implement. It simply requires putting a process into a certain partition without focusing on the emergence of Internal and External Fragmentation.

2. **Little OS overhead:**
   Processing of Fixed Partitioning requires lesser excess and indirect computational power.

Disadvantages of Fixed Partitioning –

1. **Internal Fragmentation:**
   Main memory use is inefficient. Any program, no matter how small, occupies an entire partition. This can cause internal fragmentation.

2. **External Fragmentation:**
   The total unused space (as stated above) of various partitions cannot be used to load the processes even though there is space available but not in the contiguous form (as spanning is not allowed).

3. **Limit process size:**
   Process of size greater than the size of the partition in Main Memory cannot be accommodated. The partition size cannot be varied according to the size of the incoming process size. Hence, the process size of 32MB in the above-stated example is invalid.

4. **Limitation on Degree of Multiprogramming:**
   Partitions in Main Memory are made before execution or during system configure. Main Memory is divided into a fixed number of partitions. Suppose if there are partitions in RAM and are the number of processes, then condition must be fulfilled. Number of processes greater than the number of partitions in RAM is invalid in Fixed Partitioning.

Q) 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.

Q) Explain History of Java?who invented Java?

= Java was created at Sun Microsystems, Inc., where James Gosling led a team of researchers in an effort to create a new language that would allow consumer electronic devices to communicate with each other. Work on the language began in 1991, and before long the team's focus changed to a new niche, the World Wide Web

Q) what was original name of Java?why it was renamed?

= The language was initially called Oak after an oak tree that stood outside Gosling's office. Later the project went by the name Green and was finally renamed Java, from Java coffee, a type of coffee from Indonesia.

Q) List features of Java?

= **Major Features of Java Programming Language**

- Simple.
- Object-Oriented.
- Platform Independent.
- Portable.
- Robust.
- Secure.
- Interpreted.
- Multi-Threaded

Q) List various Datatypes in Java?

= There are 8 types of Primitive data types in Java – Boolean, char, byte, int, short, long, float, and double

Q) what is difference between

System.out.print

System.out.println

System.err.print ?

= In Java System.out.println() will print to the standard out of the system you are using. On the other hand, System.err.println() will print to the standard error.

If you are using a simple Java console application, both outputs will be the same (the command line or console) but you can reconfigure the streams so that for example, System.out still prints to the console but System.err writes to a file. Also, IDEs like Eclipse show System.err in red text and System.out in black text by default.

Q) How is Java Platform independent ?

- = Java provides Platform Independence by making use of Java Byte Code. Java Byte Code or .class file is generated during the compilation of the code. This Byte Code is platform-independent and can run on any system regardless of the platform it is built upon.

Q) what is bytecode?How is it different from machine code

Both of these are codes that act as a set of instructions that help machines/ devices behave in a specified manner or perform certain operations/ tasks. The primary difference between byte code and machine code is that bytecode is an intermediate code while the machine code is the final code that the CPU processes.

Q) what is difference between Jar file & Runnable jar file?

= FYI: In simple terms, the difference between a JAR file and a Runnable JAR is that while a JAR file is a Java application which requires a command line to run, a runnable JAR file can be directly executed by double clicking it.

Q) what is difference between Runnable jar file & exe file?

= difference between jar file & Executable jar? jar file are like dead body,exe file are like living men. Jar file is the combination of compiled java classes. Executable jar file is also be combination of compiled java classes with Main class.0

Q)How is C platform dependent language

Q) what is differnce between path & classpath?

= The main difference between PATH and CLASSPATH is that Path is set for java tools in java programs like java and javac, which are used to compile your code. Whereas CLASSPATH is used by System or Application class loader to locate and load compile Java bytecodes stored in the . class file