Ques 1. Why do we need static keyword in Java Example with an example?

Ans:- In Java, the static keyword is used to define a class-level member that belongs to the class itself rather than to instances of the class. These members are shared among all instances of the class, and they can be accessed using the class name rather than an instance of the class.

**Shared Data**: Static variables (also known as class variables) are shared among all instances of the class. This allows you to maintain data that is common to all instances of the class.

**Utility Methods**: Static methods are associated with the class itself and can be called using the class name. These methods are often used for utility functions that don't need to access instance-specific data.

**Memory Efficiency**: Static members are stored in memory only once, regardless of the number of instances created. This can save memory when dealing with a large number of objects.

**Main Method**: The entry point of a Java program is the static main method.

**Example**:

class MathUtils {

static final double PI = 3.14159;

static double square(double num) {

return num \* num;

}

}

public class Main {

public static void main(String[] args) {

System.out.println("Value of PI: " + MathUtils.PI);

double result = MathUtils.square(5.0);

System.out.println("Square of 5: " + result);

}

}

Ques 2. What is class loading and does the java program actually executes?

Ans:- Class loading is a crucial step in the Java Virtual Machine's (JVM) process of executing Java programs. When you run a Java program, the program's source code is first compiled into bytecode by the Java compiler. This bytecode is then executed by the JVM.

**Loading**: The class loader loads the bytecode of a class into memory from a .class file or other sources like network locations or dynamically generated code. This step is triggered when the class is first referenced in your code.

**Initialization:** Static variables and static initializer blocks (if any) are executed. This step initializes the class's static state.

**Linking:-** It is the second step in the class loading process within the Java Virtual Machine (JVM). It involves several sub-steps that prepare the loaded bytecode for execution by resolving references, allocating memory for static variables, and ensuring bytecode validity. The linking process consists of three main components: verification, preparation, and resolution.

Ques 3 We can work a local variable as static ?

Ans:-No we can work as local variable as static.

Ques 4.Why is the static block executed before the main method in java ?

Ans:- The static blocks always execute first before the main() method in Java because the compiler stores them in memory at the time of class loading and before the object creation. Here, the compiler executes all the static blocks first, and after finishing the static block execution, it invokes the main() method.

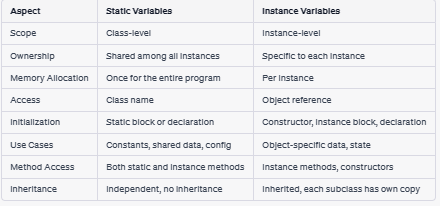
Ques 5. Why is the static method also called a class method ?

Ans:- A static method is sometimes referred to as a "class method" because it belongs to the class itself rather than an instance of the class. This is in contrast to instance methods, which operate on specific instances (objects) of the class. The term "class method" is used to emphasize that the method is associated with the class as a whole, rather than with any particular instance of the class.

Ques 6. What is the use of static block in java?

Ans:- In Java, a static block is a special block of code that is used for initializing static variables or performing one-time initialization tasks for a class. It is executed when the class is loaded by the Java Virtual Machine (JVM) and is executed before any static methods or the main method of the class are called. The static block is defined within the class and is enclosed within curly braces, just like instance initialization blocks and methods.

Ques 7. Difference between Static and Instance variables?

Ans:- 

Ques 8. Difference between static and non static members?

Ans:- 