Q1. Why do we need a static keyword in java? Explain with an example?

Ans. Static keyword mainly used for **memory management** in java. It can be applied to class, variable, and methods.This keyword is a property of class rather than an instance of a class.

Syntax: static data_type variable_name;

```
class student
{

    int i=2;//Instance Variable part of OBJECT
    static String School="D.A.V Centenary Public School";//Static Variable
part of CLASS or Non-Instance Variable
   static void study()//staticmemberfunction
   {
    System.out.println("STUDYING");
   }
}
class oneBasics
{

    public static void main(String args[])
    {    student s1 =new student();
        System.out.println(student.School);//Accessing STATIC variable
        student.study();//Static method
    }
}
Output:
D.A.V Centenary Public School;
STUDYING
```

Q2. What is class loading and how does the java program actually executes?

Ans. In java, class loading is the process of loading the **class files** into **JVM** at run-time. JVM is responsible for loading classes from various sources such as the file system, network & databases and making them available to the JVM for utilization.

3 Phases of class loading are:

1. **Loading:**In this, classloader locates the class files using fully qualified class names, it reads the class files and converts them to class objects.
2. **Linking:**The JVM performs several operations on the class object such as verifying the class file integrity, resolving symbolic references & allocating memory for the class variable.
3. **Initialization:**JVM initializes the class variable with their default values, and runs the class‘s static initialization block(if any).

Q3. Can we mark a local variable as static?

Ans. No, we cannot create a local variable as a static. Memory will be allocated to the static variable at the time of class loading, but for local and instance variable memory will be allocated at the time of object creation. So we can not declare local to static variables.

Q4. Why is the static block executed before the main method in java?
Ans. Static block is used to initialize the static data members. It gets executed before the main method because it initializes during the class loading phase of the program. It is executed only once when the class is created.

Q5. Why is a static method also called a class method?
Ans. A static method is a method that belongs to the class rather than an instance of a class(Object). We can call/invoke a static method without creating an object. That's why we call it a class method. We can access only a static variable inside a static variable. These methods are also referred to as utility methods.

Q6. What is the use of static blocks in java?
Ans. Static blocks are used to initialize the static data members. It gets executed before the main method because it initializes during the class loading phase of the program.We can create any number of static blocks, those blocks will execute from top to bottom .It is executed only once when the class is created.

Q7. Difference between static and instance variable.
Ans.

Static Variable	Instance Variable
These are also known as class variables and declared inside a class using **static keyword.**	These variable are created when an object and declared in a class but outside a method, constructor or a block
These can be accessed using a class name. **Syntax:** Class_Name.Static_variable;	These variables are accessed using a reference of class or directly inside a class. **Syntax:** object_Name.instance variable;
Only one copy of the static variable is created for the entire class.	For every instance of class we have a respective copy of the instance variable.

Q8. Difference between static and non-static members.
Ans.

Static Member	**Non-static Variable**
These members are called class variable	These members are called instance variable
Get memory in the heap area	Get memory in the method area
Value of remain same for every instance of class	Here value can be different for different instance of class

We can access static variable inside a static area	We can access both static and non-static variables in non-static areas.
Created using a static keyword	Created without using static keyword