**153. Static Members**

1. **”Static”** keyword is used to represent meta data (Data about data.)

2. Its use to represent the information of a class not about the object, though all the objects can share that data.

3. Following are the applications of static concept:

a. static variables.

b. static methods.

c. static nested class.

d. static block.

4. Information same for all objects are stored in static variables. It like class information (meta data of the class).

5. Static variables belongs to the class (Common data). They can be shared by all the objects of the class.

6. Static members can be accessed using class name without creation of an object.

7. For example:

System.out.println(Car.price);

\*Car is a class and price is static member.

8. Static members of a class are created inside the method area. They are available once the class is loaded.

9. They can be accessed by using class name or object name.

10. There will be one copy of a static member (only one instance).

11. We can change the value of a static variable but its value will be changed across instances.

12. static methods belongs to class and can be called using class and object name.

13. static methods can access only static member.

14. We cannot use **“this”** inside static methods or inside static nested classes. (this is a reference which can create ambiguity when class has multiple objects. This cannot be referenced from static context)

**154. Demo Static Members.**

1. Non-static method can access static member of a class.

2. We cannot access non-static members in static method.

3. We can’t make outer class static. Only the inner classes can be defined static.

**155. Static Blocks**

1. Static block is a block of instructions.

2. We can access only static members of class.

3. It will be executed after the class is loaded.

4. The static block is used to initialize static things. (This is not commonly used.)

**156. Demo: Static Blocks.**

1. Static block are executed first when the class is loaded.

**157. Final Keyword.**

1. Following are the applications of Final keyword:

a. Final variable.

b. Final method.

c. Final class.

2. Final variables are constant. We can’t modify the value of Final variables.

3. Final variables are written in capital letters.

4. There are three ways to initialize the final variables:

a. During the declaration.

b. Using static block/ method.

c. Constructor.

5. We cannot override final methods.

6. We cannot inherit final class.

7. Concept is used to restrict modification of methods and classes if needed.

**158. Demo: Final Keyword.**

1. We can declare final variable in method without initialization but we cannot declare final variable in class without initialization.

2. We can declare the final variable and then initialize it in a instance block.

3. We cannot override the final method. We can restrict runtime polymorphism.

4. Final class cannot be inherited or extended.

**159. Singleton Class**

1. A class such that only one object is created. If object of that class is used then that same class must be used.

2. We create singleton class using logic of static.

3. steps:

1. Make the class constructor private.

2. Create a public static function to get instance of that object.

**160. Student Challenge: Student Roll Number.**

1. Q1: Create a method and constructor so that we can generate the roll number, account number and customer number.

Student: Roll format: Uni-2020-1

Account

Cutomer

2. There layers of abstraction.

3. There class “Date” in java.util.package.

For example:

Date d = new Date();

d.getYear()+1900

d.getYear() is deprecated.

4.

5. We can define static block inside the class.

6. We can have static method implementation in interface.

7. We can overload the static methods.

8. We can’t override the static methods.