Q.3 Uses of Static and Final Classes and Methods

Static keyword can be used with class, variable, method and block. Static members belong to the class instead of a specific instance, this means if you make a member static, you can access it without object.

**static variable:**

A static variable is common to all the instances (or objects) of the class because it is a class level variable. In other words you can say that only a single copy of static variable is created and shared among all the instances of the class. Memory allocation for such variables only happens once when the class is loaded in the memory.  
Few Important Points:

Static variables are also known as Class Variables.

Unlike ****non-static variables****, such variables can be accessed directly in static and non-static methods.

**Static Methods:**

It can access class variables(static variables) without using object(instance) of the class, however non-static methods and non-static variables can only be accessed using objects.  
Static methods can be accessed directly in static and non-static methods.  
****Syntax:****  
Static keyword followed by return type, followed by method name.

**static return\_type method\_name();**

**Static Class:**

A class can be made ****static**** only if it is a nested class.

1. Nested static class doesn’t need reference of Outer class
2. A static class cannot access non-static members of the Outer class

Example:

class JavaExample{

static int i = 100;

static String s = "Beginnersbook";

//Static method

static void display()

{

System.out.println("i:"+i);

System.out.println("i:"+s);

}

//non-static method

void funcn()

{

//Static method called in non-static method

display();

}

//static method

public static void main(String args[])

{

JavaExample obj = new JavaExample();

//You need to have object to call this non-static method

obj.funcn();

//Static method called in another static method

display();

}

}

**Final classes and methods:**

## **final variable:**

## final variables are nothing but constants. We cannot change the value of a final variable once it is initialized.

### **Blank final variable**

### A final variable that is not initialized at the time of declaration is known as **blank final variable**. We **must initialize the blank final variable in constructor** of the class otherwise it will throw a compilation error.

 A static final variable that is not initialized during declaration can only be initialized in [static block](https://beginnersbook.com/2013/04/java-static-class-block-methods-variables/" \o "Java – Static Class, Block, Methods and Variables" \t "https://beginnersbook.com/2014/07/final-keyword-java-final-variable-method-class/_blank).

**2) final method**

A final method cannot be overridden. Which means even though a sub class can call the final method of parent class without any issues but it cannot override it.

## **3) final class**

We cannot extend a final class.

Example:

class Demo{

//Blank final variable

final int MAX\_VALUE;

Demo(){

//It must be initialized in constructor

MAX\_VALUE=100;

}

void myMethod(){

System.out.println(MAX\_VALUE);

}

public static void main(String args[]){

Demo obj=new Demo();

obj.myMethod();

}

}