Explain static control flow?

Great one 👍. Let’s carefully understand **Static Control Flow (SCF)** in Java.

**🔹 What is Static Control Flow?**

Static control flow = **the order in which static variables and static blocks are executed when a class is loaded**.

⚡ It happens **only once** per class, when the class is first loaded into memory by the JVM (before any object is created).

**🔹 Order of Static Control Flow**

When JVM loads a class:

1. **Identify static variables** (default values assigned first).
2. **Execute static variable assignments & static blocks** in the order they appear (top to bottom).
3. After SCF finishes, the class is ready, and then the main() method runs (if present).

**🔹 Example 1: Simple**

class Test {

static int a = 10; // step 1

static {

System.out.println("Static block");

System.out.println("a = " + a);

}

public static void main(String[] args) {

System.out.println("Main method");

}

}

**Output:**

Static block

a = 10

Main method

👉 First, a is assigned, then static block runs, then main method runs.

**🔹 Example 2: Multiple static members**

class Demo {

static int a = 10;

static int b;

static {

System.out.println("Static block 1");

b = a + 20;

}

static {

System.out.println("Static block 2");

}

public static void main(String[] args) {

System.out.println("a = " + a);

System.out.println("b = " + b);

}

}

**Output:**

Static block 1

Static block 2

a = 10

b = 30

**🔹 Example 3: Forward reference problem**

class Test {

static {

// ❌ Error: illegal forward reference

// System.out.println(x);

}

static int x = 100;

}

👉 Rule:

* You cannot use a static variable **before its declaration** (in SCF).
* But you can assign to it.

✅ Allowed:

static {

x = 200; // allowed (assignment is fine)

}

static int x = 100;

**🔹 Key Points**

* Static blocks and static variables run **before main()**.
* They run **once per class**, when class is loaded.
* Execution order = **top-to-bottom order in the source file**.
* Cannot use a static variable before it is declared (forward reference).

✅ **Summary in 1 line:**  
Static Control Flow = JVM loads class → assigns default values → executes static variable initializations & static blocks in order → finally calls main().

Would you like me to also explain **Instance Control Flow** side-by-side with Static Control Flow? That’s usually the next interview question after SCF.

**You said:**

if any other static method is there?

**ChatGPT said:**

Ah 👍 you’re asking: *In static control flow, if there are other static methods in the class, what happens?*

Good question 👌 Let’s go step by step.

**🔹 Rule**

* **Static methods** (other than main) are just **stored in the method area** when the class is loaded.
* They are **NOT executed automatically** during static control flow.
* They only run if **called explicitly** (from a static block, another method, or main).

**🔹 Example 1: Static method but never called**

class Test {

static {

System.out.println("Static block");

}

static void greet() {

System.out.println("Hello from static method");

}

public static void main(String[] args) {

System.out.println("Main method");

}

}

**Output:**

Static block

Main method

👉 Notice: greet() never runs, because we didn’t call it.

**🔹 Example 2: Call static method inside static block**

class Test {

static {

System.out.println("Static block");

greet(); // calling static method

}

static void greet() {

System.out.println("Hello from static method");

}

public static void main(String[] args) {

System.out.println("Main method");

}

}

**Output:**

Static block

Hello from static method

Main method

👉 Here, the static method ran because it was **explicitly called** inside the static block.

**🔹 Example 3: Call from main**

class Test {

static void greet() {

System.out.println("Hello from static method");

}

public static void main(String[] args) {

System.out.println("Main method");

greet(); // calling static method

}

}

**Output:**

Main method

Hello from static method

**🔹 Key Takeaway**

* **Static variables & static blocks** → executed automatically in static control flow (when class is loaded).
* **Other static methods** → only executed if explicitly called.
* main() is just another static method — but JVM calls it automatically as the program entry point.

✅ **Summary in one line:**  
In static control flow, static variables and blocks run automatically, but **other static methods run only when you call them**.

## Instance Flow Example 1: Simple ICF

class Test {

int a = 10; // instance variable

{ // instance block

System.out.println("Instance block: a = " + a);

}

Test() {

System.out.println("Constructor executed");

}

public static void main(String[] args) {

System.out.println("Main method");

Test t1 = new Test();

Test t2 = new Test();

}

}

**Output:**

Main method

Instance block: a = 10

Constructor executed

Instance block: a = 10

Constructor executed