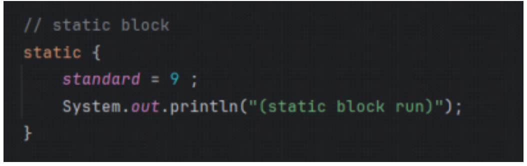
Constructor and Static Block

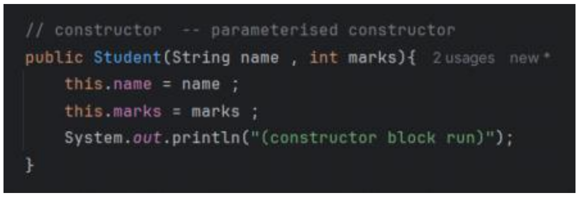
Monday, August 18, 2025 4:01 PM

1. Static Block

- A **static block** in Java is a block of code inside a class, wrapped in static { ... }.
- It runs **only once** when the class is **loaded into memory** (before any object is created and even before main() runs, if the class is used).
- Typically used for **initializing static variables**.
- 

```
// static block
static {
    standard = 9 ;
    System.out.println("(static block run)");
}
```

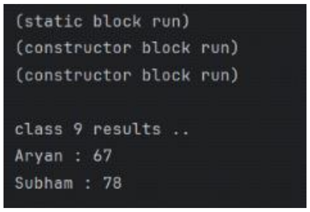
2. Constructor Block (Instance Initializer Block)

- A **constructor block** is a block inside a class (not static, just { ... }).
- It runs **every time an object is created**, **before the constructor**.
- Used for **common code** that should run before all constructors.
- 

```
// constructor  -- parameterised constructor
public Student(String name , int marks){
    this.name = name ;
    this.marks = marks ;
    System.out.println("(constructor block run)");
}
```

Notes :

1. Object creation has two steps
    a. Class loading
        - This step includes creation of .class file by JVM
    b. Object instantiation
2. Class loading is done once no matter how many objects are created
3. Static block is run in class loading step thus it run only one time
4. Constructor block is run in the object instantiation step thus it run every time an object is created
    a. Eg.



```
(static block run)
(constructor block run)
(constructor block run)

class 9 results ..
Aryan : 67
Subham : 78
```

5. The class loading step can be done explicitly
    a. Use ..
        Class.forName(className : "") ;
    b. It throws an exception and to avoid it we must do after the parenthesis of main method
        Throws classNotFoundException