



# INTRODUCTION TO PROGRAMMING LANGUAGE (JAVA)

Fariha Zahin

Lecturer

CSE, Southeast University

## What is the **static** keyword?

- A keyword used for memory management in Java.
- It is used for creating class-level members, meaning they belong to the class rather than an instance.

## Types of Static Members



## Static Variables

**Definition:** A variable declared with the static keyword is shared among all instances of the class.

### Characteristics:

- One copy of the static variable is shared across all objects of the class.
- The static variable is initialized only once when the class is loaded into memory.
- Variables declared with the static keyword can be called without creating an object of the class.

```
class Student{  
    int id;  
    static String courseName="java";  
  
    Student(int id){  
        this.id=id;  
    }  
}
```

```
public static void main(String[] args) {  
    System.out.println(Student.courseName);  
}
```

```
com.mycompany.lab3.Lab3 >  
Run - Run (Lab3) ×  
· java
```

**Brainstorming 1** : What will be the value of count if 3 objects of Counter class is created?

```
class Counter {  
    static int count = 0;  
  
    Counter() {  
        count++;  
    }  
}
```

## Static Methods

**Definition:** Methods declared with the static keyword can be called without creating an object of the class.

### Characteristics:

- A static method can only access other static members (variables and methods).
- `this` and `super` keywords cannot be used here.

```
11  class Student{
12      int id;
13      static String courseName="java";
14
15      Student(int id){
16          System.out.println(courseName);
17          this.id=id;
18      }
19
20      static displayInfo()
21      {
22          System.out.println(id);
23          System.out.println(courseName);
24      }
```



**Brainstorming 2 : why main method is declared as static?**



## Static Blocks

**Definition:** A static block is used to initialize static variables or perform setup when the class is loaded.

### Characteristics:

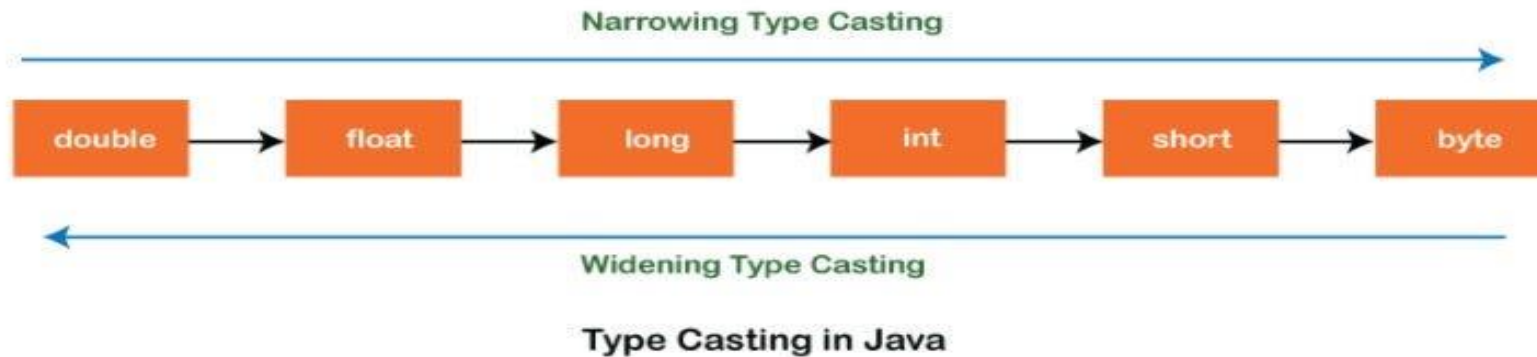
- Executed once when the class is loaded into memory.
- Can throw exceptions.
- Executed before main method.

```
class StaticBlockExample {  
    static {  
        System.out.println("Static block executed");  
    }  
}
```

## Type Casting

**Type Casting** is the process of converting one data type into another. It is of two types:

1. **Implicit Type Casting (Widening)**
2. **Explicit Type Casting (Narrowing)**



### 1. Implicit Type Casting (Widening)

**Definition:** Implicit type casting occurs when a smaller data type is automatically converted to a larger data type. This happens without the need for explicit casting, and it does not result in data loss.

.



## 2. Explicit Type Casting (Narrowing)

**Definition:** Explicit type casting occurs when a larger data type is manually converted to a smaller data type. This requires the programmer to specify the type they want to cast the data to.

### How it works:

- When narrowing, there is a risk of **data loss** if the value doesn't fit in the target type. The programmer must be cautious when performing explicit casting.
- **Syntax for explicit casting:** *(targetType) value;*

```
int x = 100;  
long y = x; // Implicit casting from int to long  
float z = y; // Implicit casting from long to float
```

```
double a = 9.78;  
int b = (int) a; // Explicit casting from double to int  
System.out.println(b); // Output: 9 (data loss occurs)
```