



CSN-103: Fundamentals of Object Oriented Programming

Instructor: Dr. Rahul Thakur

Assistant Professor, Computer Science and Engineering, IIT Roorkee



- A self-contained block of statements that perform a coherent task of some kind
 - Also known as **Function** (*in different programming languages*)
 - Block of statements are executed **only** when method it is called
- Examples:
 - `main()`
 - `println()`
 - `nextInt()`, `nextFloat()`, `nextLine()`...
 - `clone()`
 - `length()`, `charAt()`, `toArray()`, `equals()` ...

Creating and Calling a Method

- Creating a method
 - Must be declared within a class
 - General Form

```
modifier returnType nameOfMethod (Parameter List)
{
    // method body
}
```

- Calling a method
 - Method's name followed by two parentheses () and a semicolon ;

```
nameOfMethod (Argument List);
```

Parameters, Arguments, and Return Type

- Parameters vs. Arguments

```
static int Add(int k, int l)
{
    int s = k+l;
    return s;
}
```

Add(int i, int j);

Add(3, 4);

- Return Statement and Return Type

- **return** statement:

- Immediately transfers the control back to the calling program
- Return a **value** to the calling program

- Return **type** should be same as the type of **value** returned

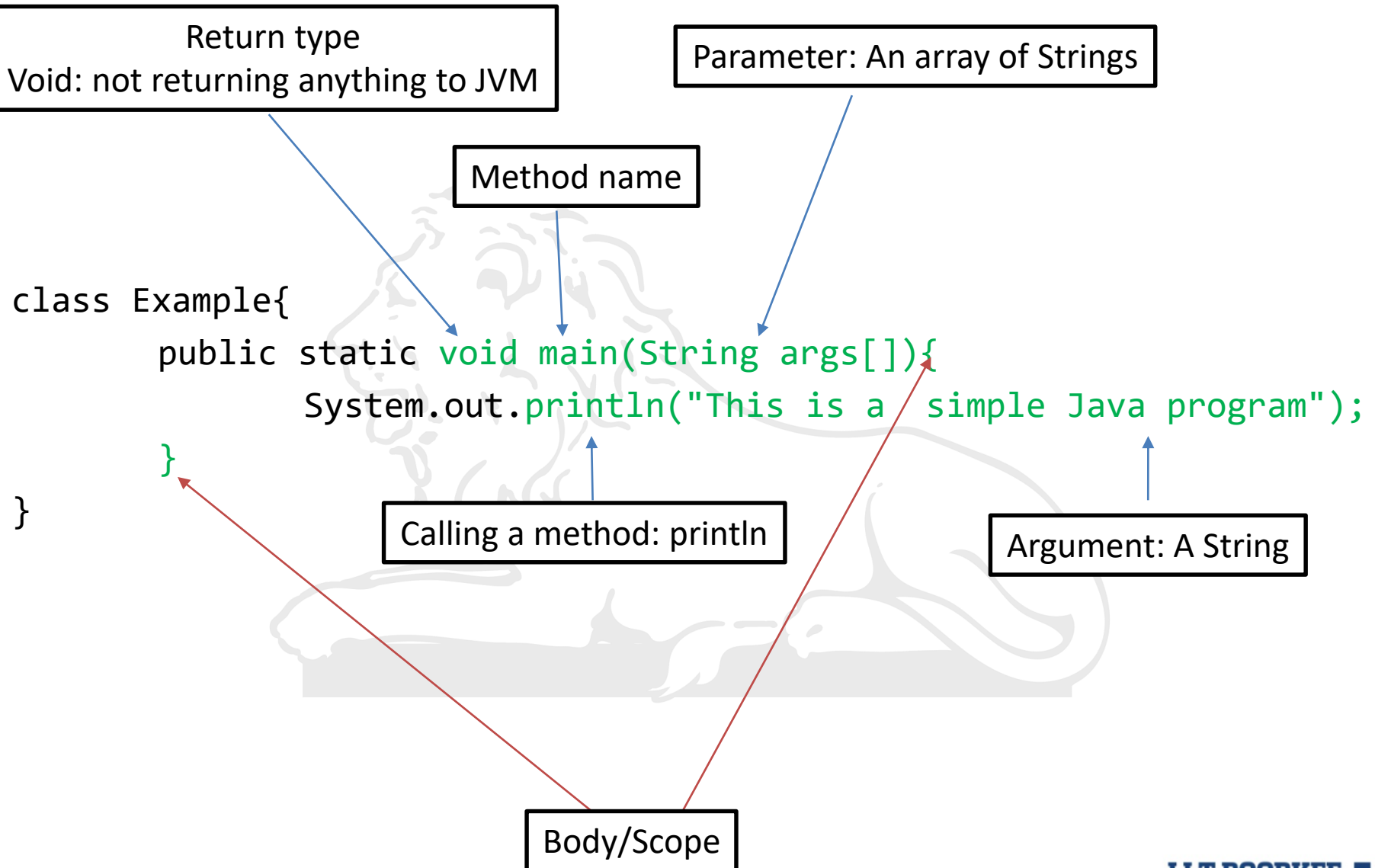
Call by Value and Call by Reference

- Parameters passing method
 - By Value: Values of arguments are copied to the parameters
 - Changes made inside functions are not reflected in arguments
 - All simple types variable are passes as Call By Value
 - By Reference: Both the arguments and parameters refer to same locations
 - Changes made inside functions are reflected in arguments
 - All *Objects* are passes as Call By Reference
- Scope of variables
 - Accessibility of variables (Lifetime)
 - Variables declared inside a method **can't be** accessed outside the method

Advantages

- Advantages of using a method
 - Avoids rewriting the same code over and over
 - Only runs when it is called
 - Easier to write/debug programs
 - Abstraction
 - Modularity
- **Procedural Programming**

Revisiting the main() method



Command-Line Arguments

- Pass information into to the main() when you run it
 - **Command-line arguments**

```
class CommandLine{  
    public static void main(String args[]) {  
        for(int i=0; i<args.length; i++)  
            System.out.println("args[" + i + "]:"+ args[i]);  
    }  
}
```

- Stored as **strings** in a String array passed as parameter of **main()**

Parsing Strings

- Java uses primitive types (simple types)
 - Such as int or double
 - Holds and manipulates the basic data types
- Primitive type provides performance benefit
 - Java provides equivalent objects for these primitive types (type wrappers)
 - Unacceptable overhead even for simplest of calculations
- Type Wrappers or **Wrapper Classes**
 - Double, Float
 - Integer, Short, Byte, Long
 - Character
 - Boolean