



CSN-103: Fundamentals of Object Oriented Programming

Instructor: Dr. Rahul Thakur

Assistant Professor, Computer Science and Engineering, IIT Roorkee

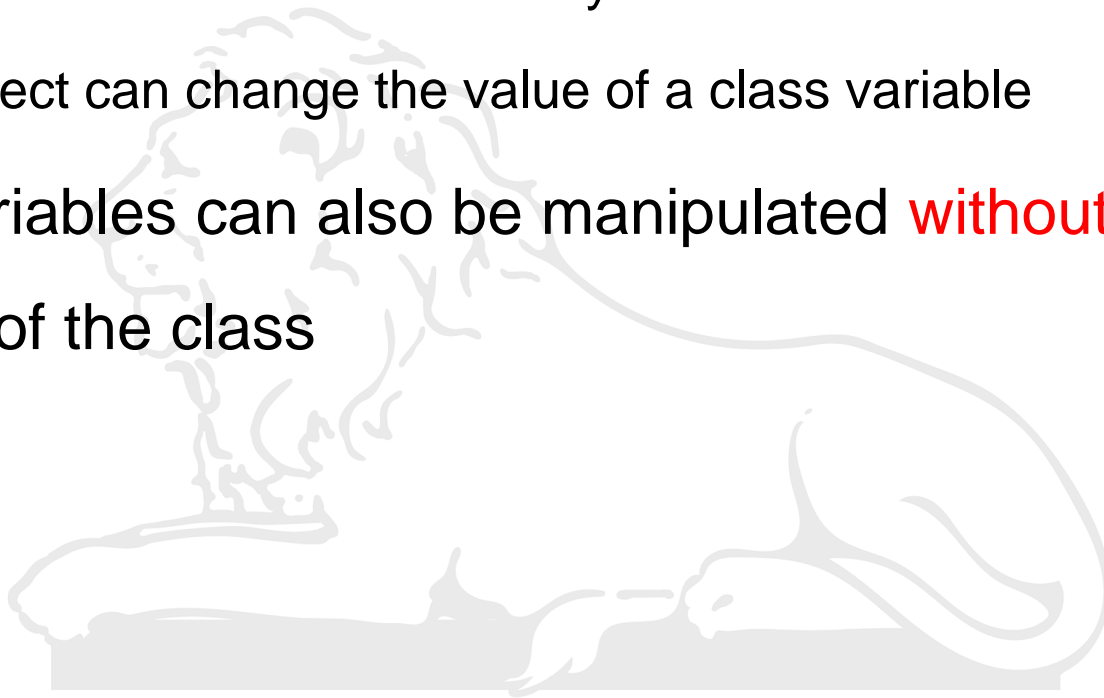


static Keyword

- When objects are created from the same class
 - Each have their **own distinct copies** of *instance variables*
- What if you want to have a variable that is **common** to all objects
- **static** variable
 - Preceding variable declaration with the keyword **static**
- **static** variables are associated with **the class**, rather than with any **object**

static Keyword

- Every instance of the class **shares** the static variable(s)
 - Just one fixed location in memory
 - Any object can change the value of a class variable
- **static** variables can also be manipulated **without** creating an instance of the class



static Keyword

- Methods can also be declared as **static**
- public **static** void **main()**
 - **main()** can be called without creating an object

static Method
Called without creating an object of Example class

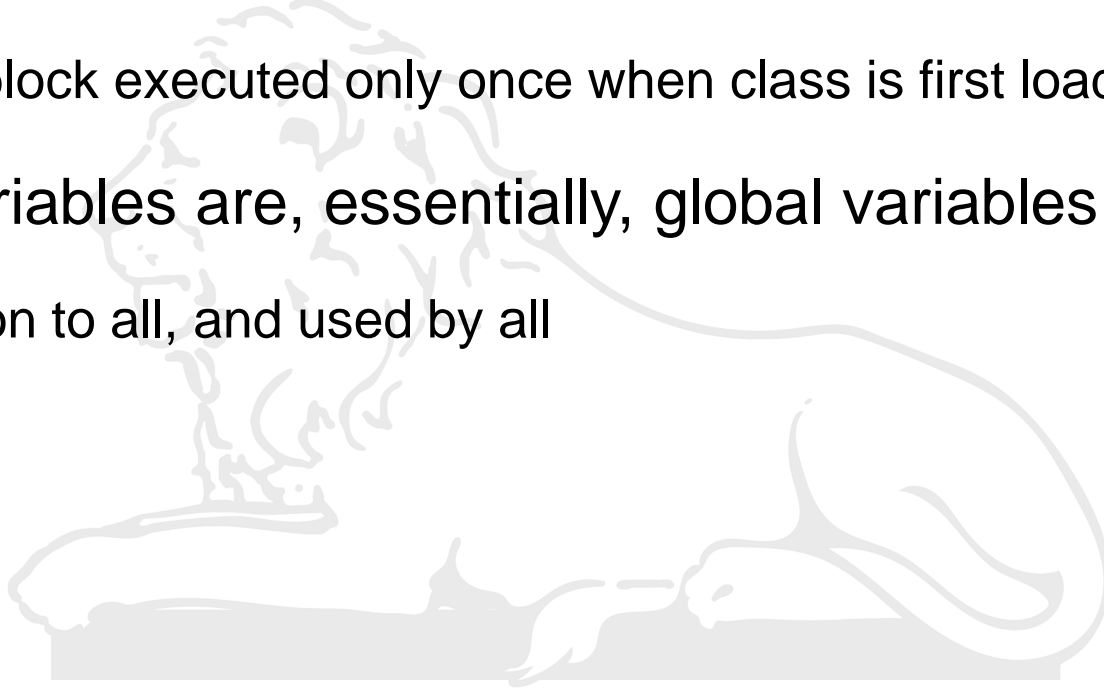
```
class Example{  
    public static void main(String args[]){  
        System.out.println("This is a simple Java program");  
    }  
}
```

static Object

- **static** methods have several restrictions
 - They can **only access static data**
 - They can call only other static methods
 - They can't refer to **this** or **super**(Inheritance)

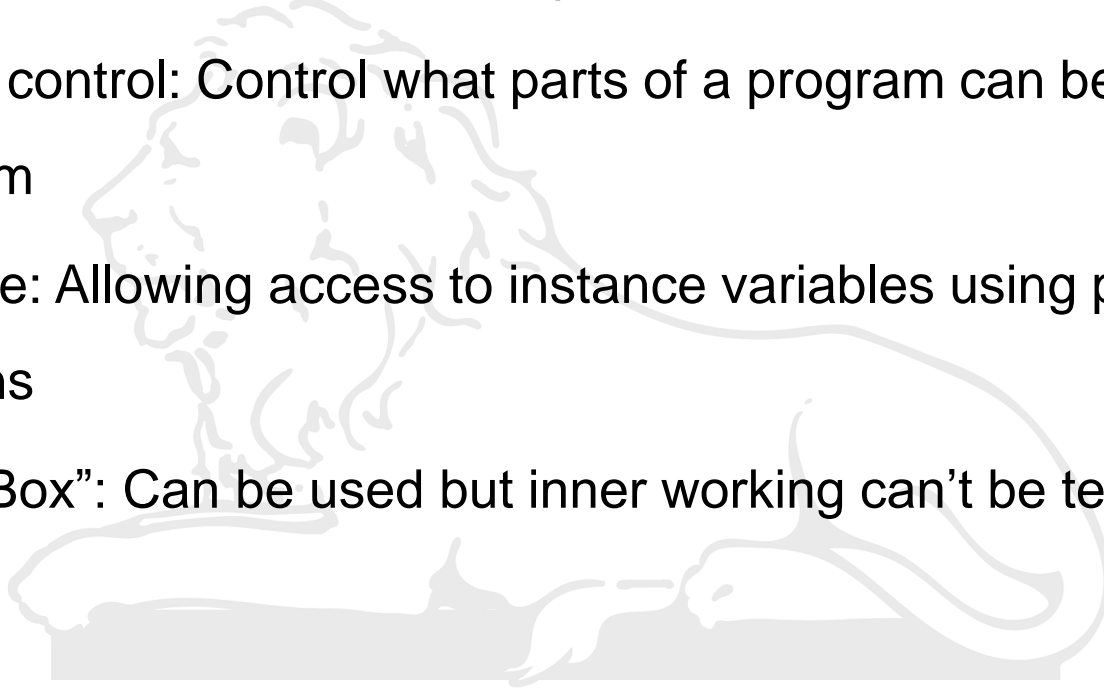
static Keyword

- If you wish to initialize **static** variables:
 - Declare a **static** block
 - **static** block executed only once when class is first loaded
- **static** variables are, essentially, global variables
 - Common to all, and used by all



- Encapsulation

- Links **data** with **code** that manipulates it
- Access control: Control what parts of a program can be access and by whom
- Example: Allowing access to instance variables using predefined functions
- “Black Box”: Can be used but inner working can’t be tempered



Access Control

- Control access of a **member** by the **Access Specifier**
 - Public
 - Private
 - Protected
 - *Package-Private* (no explicit modifier): **Default**
- Access control can also be done at the **class** level
 - Public
 - *Package-Private* (no explicit modifier): Default

Access Control for Class Members

- Member access specifier

- **Public:** Member can be accessed by any other code

- Revisiting the main() method

```
class Example{  
    public static void main(String args[]){  
        System.out.println("This is a simple Java program");  
    }  
}
```

public Method
Called by a code (Java Runtime System) outside the program

- **Private:** Member can be accessed by other members of **its class**

- **Protected:** Member can only be accessed within its own package

- + *by a subclass of its class in another package* ← (Inheritance)

Access Control



Access Levels				
Modifier	Class	Package	Subclass	World
public	Y	Y	Y	Y
protected	Y	Y	Y	N
<i>no modifier</i>	Y	Y	N	N
private	Y	N	N	N

