

Static

Sungchul Lee

Learning Object

- Instance vs. Static

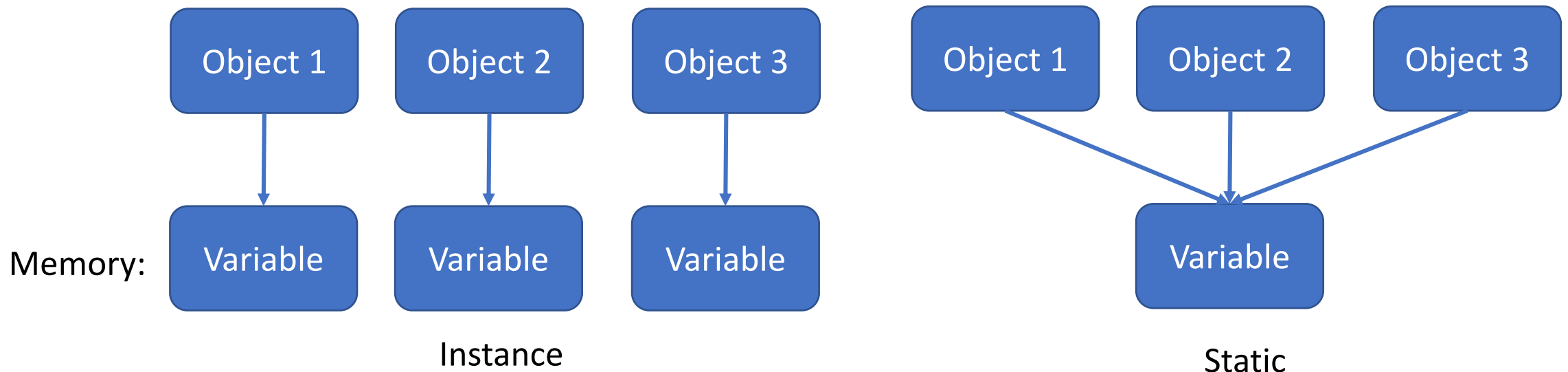
- Static

 - ❑ Static Variable

 - ❑ Static Methods

Instance vs. Static

- A variable or method that is **dependent on a specific instance** of the class should be an **instance** variable or method.
- A variable or method that is **not dependent on a specific instance** of the class should be a **static** variable or method.



Static

- Syntax:
static <type> <variable name>
- Static keyword
 - ❑ Already used in main method
- Upload the variables and Methods on memory when starting the program
 - ❑ Class Variables could be static
 - ❑ Class Methods could be static
- Save usage of memory and Reduce code

Static Variable

- If you declare a variable static, it is a variable shared by all instances of the **class**.
 - e.g. `static int a;` (class variable)
- **A static variable belongs to a class as a whole and not to an individual object.**
- A static variable is shared by all the objects of its class.
- **Static variables** store values for the variables in a common memory location, all objects of the same class are affected if one object changes the value of a static variable.

Static Methods

➤ Syntax:

`<modifier> static <return type> <method name> (params) {}`

- Java supports **static methods** as well as static variables. Static methods can be called without creating an instance of the class
 - ❑ Sometimes you need a method that has no relation to an object of any kind. In these case, you can define the method as static.
- If you declare a method static, it can be called independently of any object through its class name.

Static Methods – cont.

- When you define a static method, the method is still a member of a class, since you define it in a class, but the method can be **invoked without using any object.**
 - ❑ **Direct use the method without declaring object**
- A static method cannot reference an instance variable.
- A static method can call only a static method.
- Main is a static method.
- A non-static method can reference any variable or call any method within its class.

Practice

1. Make a new project (Reference: Create Project and Class File)
 - ☐Project name: Static
2. Create a new Class File
 - ☐Class name: Main
 - ☐Class name: Static
3. Coding:

Practice – Code (Car)

```
public class Car {  
    String name;  
    static String manufacturer = "Transformers";  
    public Car (String name){  
        this.name = name;  
    }  
    public String getName() {  
        return name;  
    }  
    public void setName(String name) {  
        this.name = name;  
    }  
    public static String getManufacturer() {  
        return manufacturer;  
    }  
    public static void setManufacturer(String manufacturer) {  
        Car.manufacturer = manufacturer;  
    }  
}
```



No this keyword

Practice – Code (Main)

```
public class Static {  
    public static void main(String[] args) {  
  
        Car bumblebee= new Car("Bumblebee"); //object  
        Car optimus= new Car("Optimus Prime"); //object  
        System.out.println("Original Manufacturer:");  
        System.out.println(bumblebee.getManufacturer());  
        System.out.println(optimus.getManufacturer());  
        bumblebee.setName("Bumblebee 2");  
        bumblebee.setManufacturer("Dr.Lee Company");  
    }  
}
```

Practice – Code (Main) – cont.

When change static variable in a object (Same Class), all other Object's static variable also are changed.

```
System.out.println("After sold to Dr.Lee");  
System.out.println(bumblebee.getName());  
System.out.println(bumblebee.getManufacturer());  
System.out.println(optimus.getName());  
System.out.println(optimus.getManufacturer());  
  
Car.setManufacturer("DreamWorks"); //Using Car class  
System.out.println("After sold to DreamWorks");  
System.out.println(bumblebee.getName());  
System.out.println(bumblebee.getManufacturer());  
System.out.println(optimus.getName());  
System.out.println(optimus.getManufacturer());  
}  
}
```

Practice – Code and Result

Car.java

```
1 public class Car {
2     String name;
3     static String manufacturer = "Transformers";
4
5     public Car (String name){
6         this.name = name;
7     }
8     public String getName() {
9         return name;
10    }
11    public void setName(String name) {
12        this.name = name;
13    }
14    public static String getManufacturer() {
15        return manufacturer;
16    }
17    public static void setManufacturer(String manufacturer) {
18        Car.manufacturer = manufacturer;
19    }
20 }
21 }
```

Static.java

```
1 public class Static {
2     public static void main(String[] args) {
3         Car bumblebee= new Car("Bumblebee"); //object
4         Car optimus= new Car("Optimus Prime"); //object
5         System.out.println("Original Manufacturer:");
6         System.out.println(bumblebee.getManufacturer());
7         System.out.println(optimus.getManufacturer());
8         bumblebee.setName("Bumblebee 2");
9         bumblebee.setManufacturer("Dr.Lee Company");
10
11         System.out.println("After sold to Dr.Lee");
12         System.out.println(bumblebee.getName());
13         System.out.println(bumblebee.getManufacturer());
14         System.out.println(optimus.getName());
15         System.out.println(optimus.getManufacturer());
16
17         Car.setManufacturer("DreamWorks"); //Using Car class
18         System.out.println("After sold to DreamWorks");
19         System.out.println(bumblebee.getName());
20         System.out.println(bumblebee.getManufacturer());
21         System.out.println(optimus.getName());
22         System.out.println(optimus.getManufacturer());
23     }
24 }
```

Problems Javadoc Declaration Console

```
<terminated> Static [Java Application] C:\Progr
Original Manufacturer:
Transformers
Transformers
After sold to Dr.Lee
Bumblebee 2
Dr.Lee Company
Optimus Prime
Dr.Lee Company
After sold to DreamWorks
Bumblebee 2
DreamWorks
Optimus Prime
DreamWorks
```



Result

Summary

➤ Instance vs. Static

➤ Static

- ❑ Static Variable

- ❑ Static Methods

```
String name;  
static String manufacturer = "Transformers";  
  
public Car (String name){  
    this.name = name;  
}  
public String getName() {  
    return name;  
}  
public void setName(String name) {  
    this.name = name;  
}  
public static String getManufacturer() {  
    return manufacturer;  
}  
public static void setManufacturer(String manufacturer) {  
    Car.manufacturer = manufacturer;  
}
```

