MODULE 1: PROGRAMMING IN JAVA

Week 7

Most Essential Learning Competencies: *At the end of the course, you must be able to:*

1. Apply static keywords to methods and fields

2. Write overload method

3. Explore access modifiers

Reading Activity

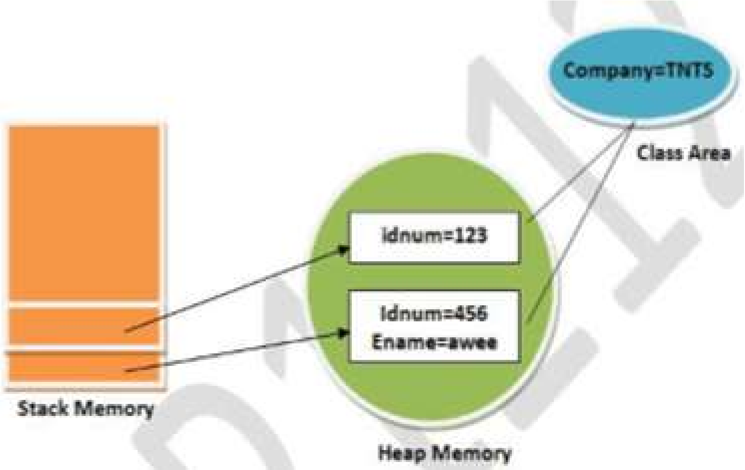


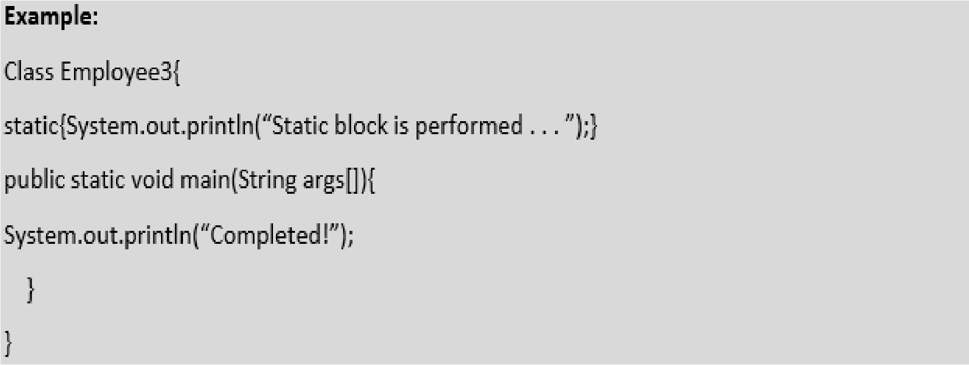
Figure 7.1 Applying Static Keywords to Methods and Fields

Primarily in memory management, the static keyword is used in Java. Static keyword can be applied methods, variables, nested classes, and blocks. The static can be a variable, method, and block.

Keyword – Word with a predefined meaning in Java programming language syntax.

static – Keyword denotes that a member variable or method can be accessed without requiring an instantiation of the class to which it belongs.

1. A **static variable** can be utilized to indicate the common property of all objects. It acquires memory only at the time of class loading
2. When static keyword is applied with a method, it is referred to as **static method**. It belongs to the class rather than object. It can be used without the necessity of creating an instance of a class. It can retrieve static data member and can alter the value of it.
3. A **static block** is used to prepare a static data member. At the time of class loading. It is performed before the main method.



Example:

```
Class Employee3{
static{System.out.println("Static block is performed . . . ");}
public static void main(String args[]){
System.out.println("Completed!");
   }
}
```

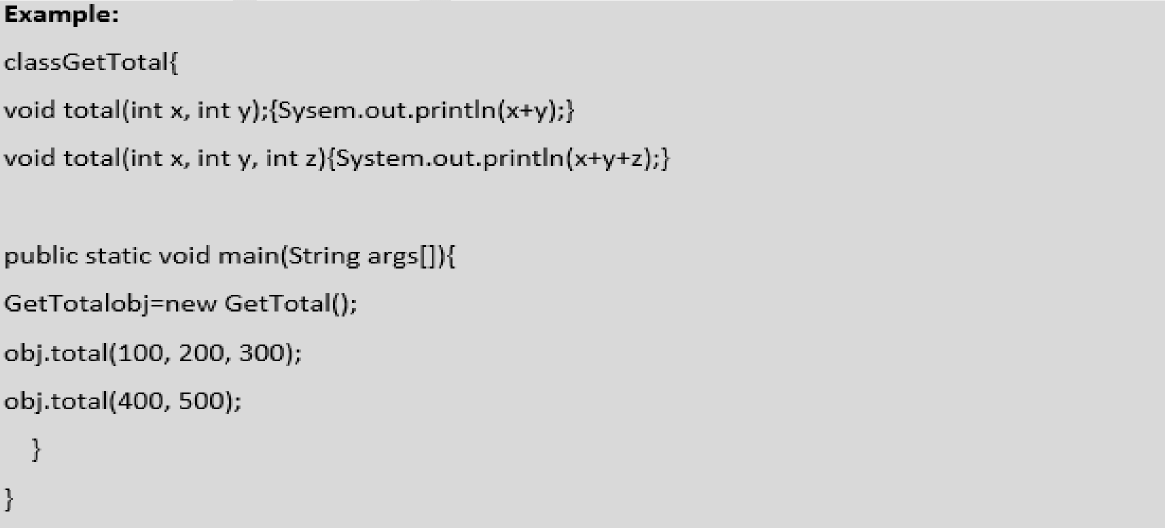
4. **Nested classes** are defined within another class.

Creating Overloading Method

It is likely to identify two or more methods within the same class that uses the same name, as long as their parameter declarations are not the same. If this is done, the methods are understood to be overloaded and the process is known as ***method overloading***. Method overloading is an example of ***polymorphism***.

1. **Overloading** - Feature that allows a class to have two or more methods having the same name, if their arguments lists are different.
2. **Polymorphism** – Ability of an object to take on many forms.

   To overload a method, you can either change the number of arguments or change the data type.



Example:

```
classGetTotal{
void total(int x, int y);{Sysem.out.println(x+y);}
void total(int x, int y, int z){System.out.println(x+y+z);}

public static void main(String args[]){
GetTotalobj=new GetTotal();
obj.total(100, 200, 300);
obj.total(400, 500);
   }
}
```

In this example, two overloaded methods are created. First total method performs addition of two numbers and second total method performs addition of three numbers.

Applying Access Modifiers

Access modifiers are used to set access levels for classes, variables, methods, and constructors. The four access levels are ***default, private, public,*** and ***protected.***

1. **Default.** Default access modifier is not declared an access modifier for a class, field, method, etc. Without any access control modifier. It can be used by any other class in the same package.

2. **Private.** When methods, variables, and constructors are declared private, they can only be accessed inside the declared class. You cannot declare class and interface as private.

3. **Public.** Declaring a class, method, constructors, and interface as public. It can be used from any other class, fields, methods, blocks declared inside a public class can be used in any class.

4. **Protected.** When variables, methods and constructors are declared protected in a superclass, they can be used by the subclasses in another package or any class within the package of the protected members' class. Such protected access modifiers cannot be applied to class and interfaces. Protected access offers the subclass a chance to access the helper method or variable while blocking a nonrelated class from using it.

Read Additional Resources

1. **Static Keyword in Java (PDF)**
   https://www.fadelk.com/files/Resources/173/YouTube/StaticKeywordPart1.pdf
2. **Method Overloading in Java**
   https://pragjyotishcollege.ac.in/wp-content/uploads/2020/04/Method-Overloading-in-Java.pdf
3. **Java Class and Inheritance**
   http://www.nyu.edu/classes/jcf/g22.2110-001_fa10/handouts/Java_Class_And_Inheritance.pdf

Watch Video Resources

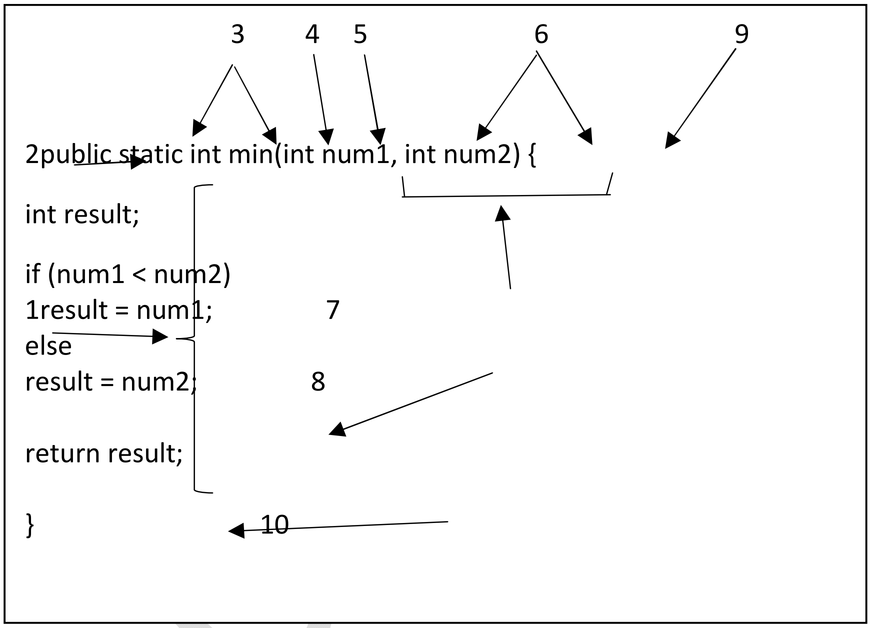
1. **Access Modifiers**
   https://youtu.be/ePj64t65G40
2. **Packages**
   https://youtu.be/l5SviD48vOQ
3. **Overloaded Method**
   https://youtu.be/LesVxjt7PoY

☑ Self-Check

Quiz 7.1

Instructions: *Write your answer on the Answer Sheet (AS) provided in this module.*

A. Enumeration. Enumerate the following part of a method structure. (1-point each)



```
2public static int min(int num1, int num2) {

int result;

if (num1 < num2)
1result = num1;
else
result = num2;

return result;

}
```

Quiz 7.2

B. Identification. Identify what is asked. (1-point each)

1. Ability of an object to take on many forms.
2. Word with a predefined meaning in Java programming language syntax.
3. Keyword denotes that a member variable or method can be accessed without requiring an instantiation of the class to which it belongs.
4. It can be utilized to indicate the common property of all objects.
5. It is used to prepare a static data member.
6. A Feature that allows a class to have two or more methods having the same name, if their arguments lists are different.
7. It is not declared an access modifier for a class, field, method, etc. Without any access control modifier.

8. When variables, methods and constructors are declared protected in a superclass, they can be used by the subclasses in another package
9. When methods, variables, and constructors are declared private, they can only be accessed inside the declared class
10. Declaring a class, method, constructors, and interface as public.

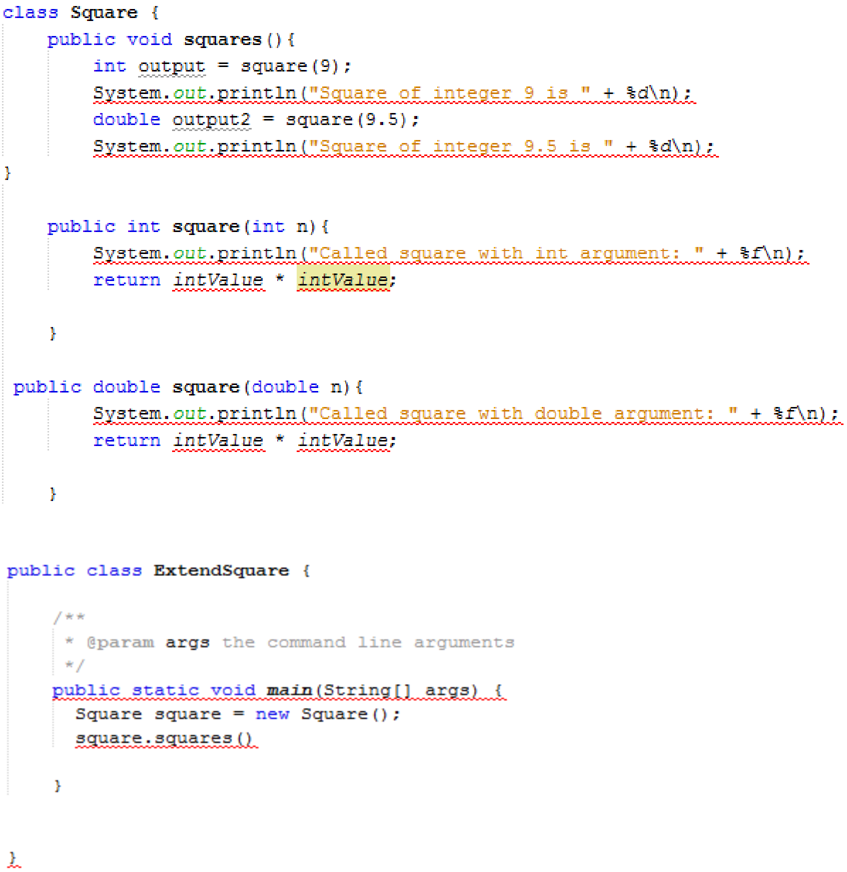
Laboratory Activity

Activity 7.1

1. Open IDE

2. Type the code below as shown

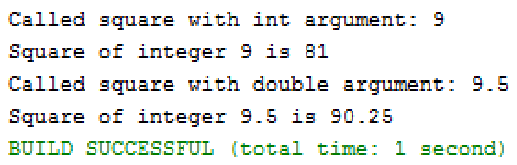
3. Save as **Square.java**



```
class Square {
    public void squares(){
        int output = square(9);
        System.out.println("Square of integer 9 is " + %d\n);
        double output2 = square(9.5);
        System.out.println("Square of integer 9.5 is " + %d\n);
}

    public int square(int n){
        System.out.println("Called square with int argument: " + %f\n);
        return intValue * intValue;

    }

 public double square(double n){
        System.out.println("Called square with double argument: " + %f\n);
        return intValue * intValue;

    }


public class ExtendSquare {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
      Square square = new Square();
      square.squares()

    }


}
```

4. If there are errors, debug the program. Compile it again. Do this repeatedly until no errors will be detected.

5. This will be the final output.



```
Called square with int argument: 9
Square of integer 9 is 81
Called square with double argument: 9.5
Square of integer 9.5 is 90.25
BUILD SUCCESSFUL (total time: 1 second)
```

Insights

Insights 7.1

Instructions: *Write your answer on the Insights Sheet (IS) provided in this module.*

We wanted to know how deep your understandings about the topics presented in the module. Your answer will be graded accordingly using the prescribed rubrics. In your own words (at least 200 words):

1. Difference between overloading and overriding in Java.(10-points)
2. Give and explain the four access modifiers.(10-points)

Internet References

1. https://www.google.com
2. https://www.fadelk.com
3. https://pragjyotishcollege.ac.in
4. https://www.youtube.com/user/isaykatsman/featured
5. https://www.youtube.com/c/CodeMonkeyCharlie/about
6. http://www.nyu.edu/
7. Phoenix Publishing House Inc.