Programming Techniques2

Section (2)

Aya Saleh Abd El_Mohsen

Method

- A **method** is a block of code which only runs when it is called. You can pass data, known as parameters, into a method.

- Methods are used to perform certain actions, and they are also known as **functions**.

Types of Method

- Standard Library Methods:

The standard library methods are built-in methods in Java that are readily available for use.

- User-defined Methods:

You can also define methods inside a class as per your wish. Such methods are called user-defined methods.

Standard Library Methods:

Categories of Built in Methods

- i) String Methods
- ii) Number Methods
- iii) Character Methods
- iv) Array Methods etc…

String Methods

1) compareTo() Method (It compares two strings, supports 3-way comparison)
Result Criteria for 3-way comparison

- Example:

```
public static void main (String [] args){
String str1 = “selenium”;
String str2 = “SELENIUM”;
String str3 = “seleniuma”;
String str4 = “selenium”;

System.out.println(str1.compareTo(str2));//Positive value
System.out.println(str1.compareTo(str3));//Negative value
System.out.println(str1.compareTo(str4));//0
}
}
```

String Methods

2) equals () Method (It compares two strings and supports 2-way comparison)

- Example:

```
public static void main (String [] args){
String str1 = “selenium”;
String str2 = “SELENIUM”;
String str3 = “selenium”;

System.out.println(str1.equals(str2));//false
System.out.println(str1.equals(str3));//true
}
```

String Methods

3) concat() Method (It concatenates two strings /Joins two strings)

- Example:

```
public static void main (String [] args){
String str1 = “Selenium”;
String str2 = “Testing”;

System.out.println(str1.concat(str2));//SeleniumTesting
System.out.println(str1 + str2);//SeleniumTesting
}
```

String Methods

4) charAt() Method (Returns a character by index position)

- Example:

```
public static void main (String [] args){
String str1 = “Selenium”;

System.out.println(str1.charAt(1));//e
}
```

String Methods

5) toUpperCase () – Converts values to Upper case)

Example:

```
public static void main (String [] args){
    String str1 = “SELENIUM”;
    String str2 = “selenium”;
    String str3 = “SELEnium”;
    String str4 = “selenium123”;

    System.out.println(str1.toUpperCase());//SELENIUM
    System.out.println(str2.toUpperCase());//SELENIUM
    System.out.println(str3.toUpperCase());//SELENIUM
    System.out.println(str4.toUpperCase());//SELENIUM123
    }
```

Try to use **toLowerCase()**

String Methods

6) endsWith() -Ends with specified suffixExample:

Example :

```
public static void main (String [] args){
String str = “Welcome to Selenium Testing”;

System.out.println(str.endsWith(“Selenium Testing”));//true
System.out.println(str.endsWith(“Testing”));//true
System.out.println(str.endsWith(“Selenium”));//false
}
```

String Methods

7) length() (returns string length)

Example :

```
public static void main (String [] args){
String str = “Selenium Testing”;
String str2 = “Selenium”;
System.out.println(str.length());//16
System.out.println(str2.length());//8
}
```

Number Methods

1) abs() -Returns absolute value

Example :

```
public static void main (String [] args){
double a =10.234;
double b =-10.784;
System.out.println(Math.abs(a));//10.234
System.out.println(Math.abs(b));//10.784
}
```

Number Methods

2) round() -It rounds the value to nearest integer

Example :

```
public static void main (String [] args){
double a =10.234;
double b =-10.784;
double c =10.51;
System.out.println(Math.round(a));//10
System.out.println(Math.round(b));//-11
System.out.println(Math.round(c));//11
}
```

Number Methods

3) min() – Returns minimum value between two numbers

Example :

```
public static void main (String [] args){
int a=10, b=20;
double c =10.234, d =10.345;
System.out.println(Math.min(a, b));//10
System.out.println(Math.min(c, d));//10.234
System.out.println(Math.min(7, 9));//7
System.out.println(Math.min(1.23, 1.234));//1.23
}
```

Try to use **max()**

Number Methods

4) random() – Generates a random number

Example :

```
public static void main (String [] args){
System.out.println(Math.random());//
}
```

Number Methods

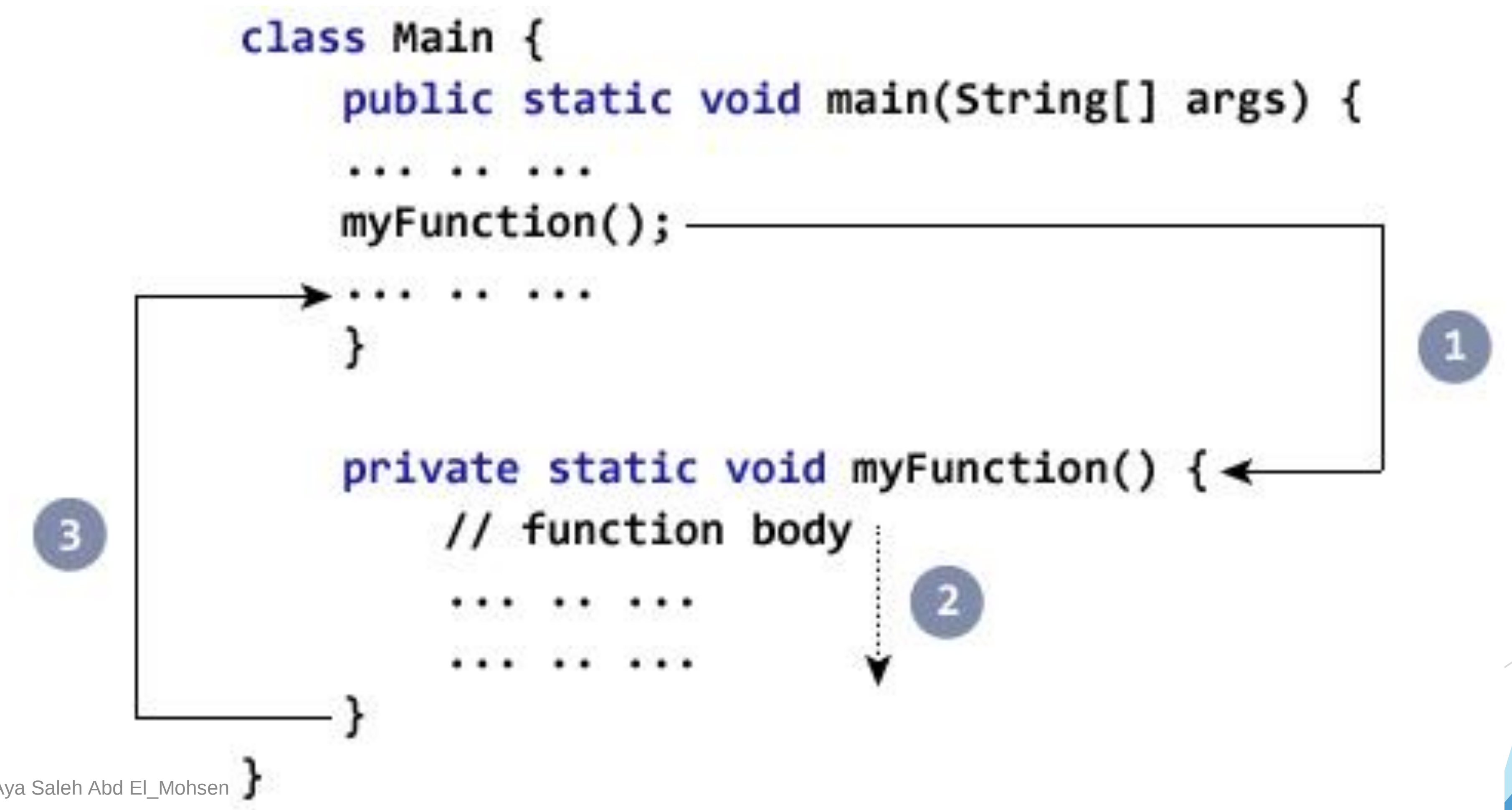
5) sqrt() – method of Math class. It returns square root of a number.

Example :

```
public class Numbers {
 public static void main(String[] args)
  {
        System.out.print("Square root of 4 is: " + Math.sqrt(4));
  }
  }
```

The output will be Square root of 4 is: 2.0

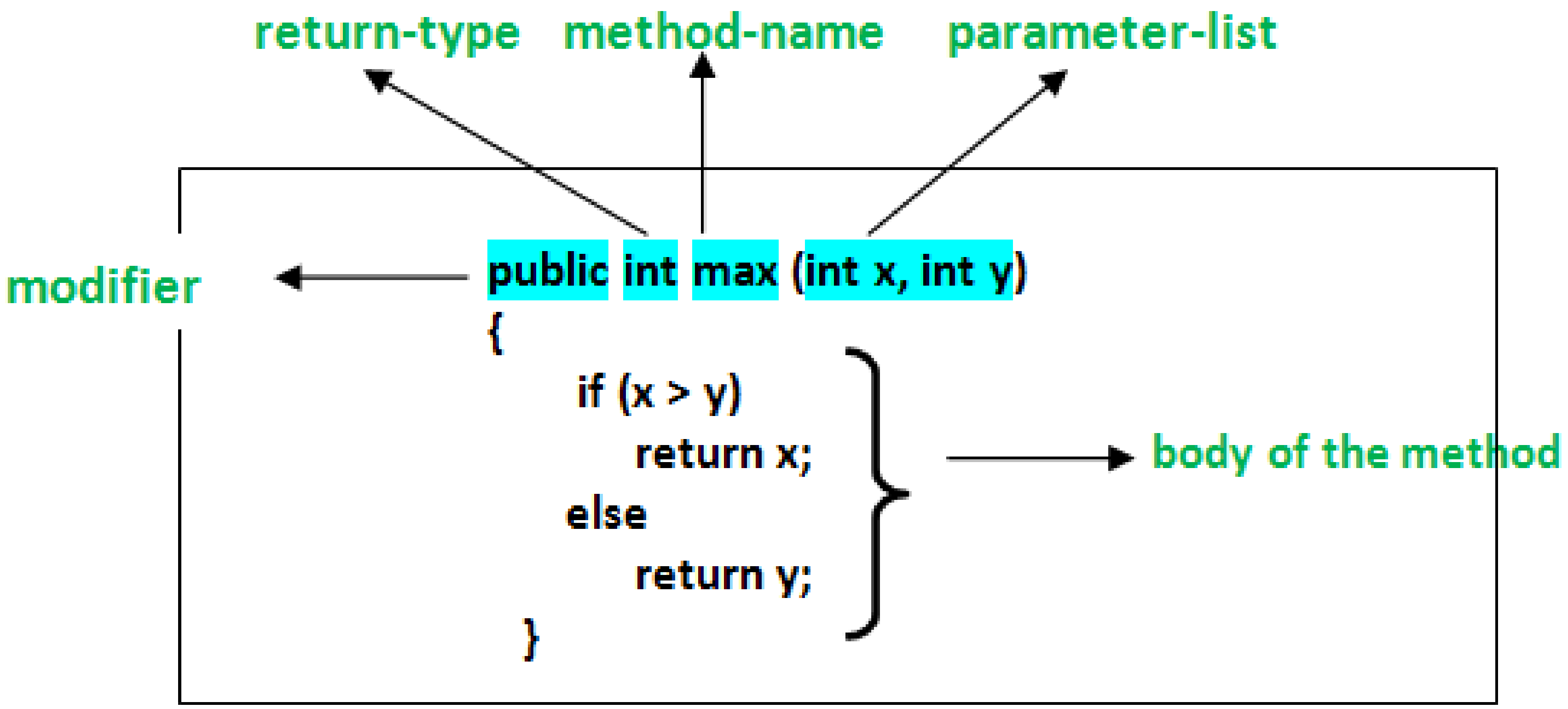
User defined Method



```
class Main {
    public static void main(String[] args) {
    ... .. ...
    myFunction();
    ... .. ...
    }

    private static void myFunction() {
        // function body
        ... .. ...
        ... .. ...
    }
}
```

User defined Method (Con…)

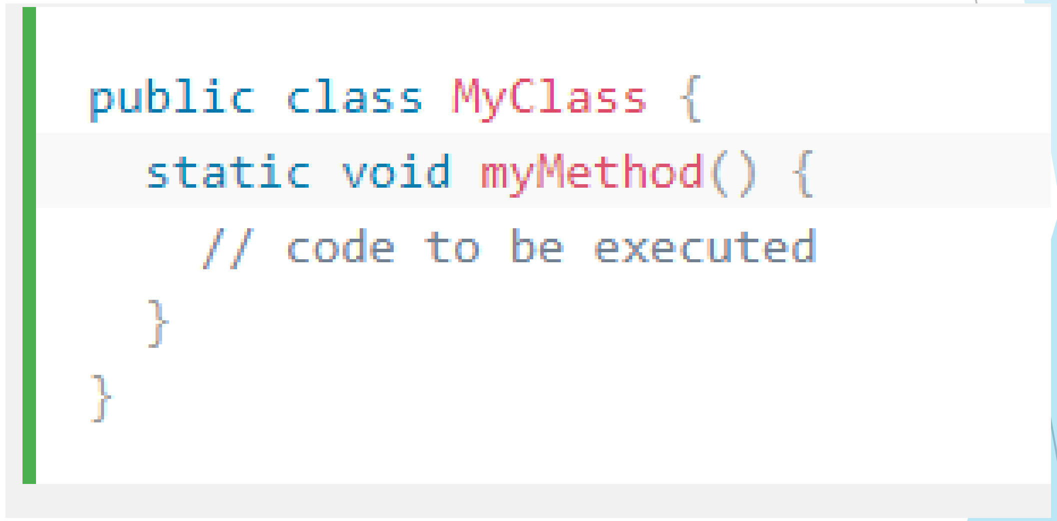


```
public int max (int x, int y)
{
      if (x > y)
          return x;
      else
          return y;
  }
```

Creating a method

- **A method must be declared within a class. It is defined with the name of the method, followed by parentheses ().**

- **Java provides some pre-defined methods, such as System.out.println(), but you can also create your own methods to perform certain actions:**



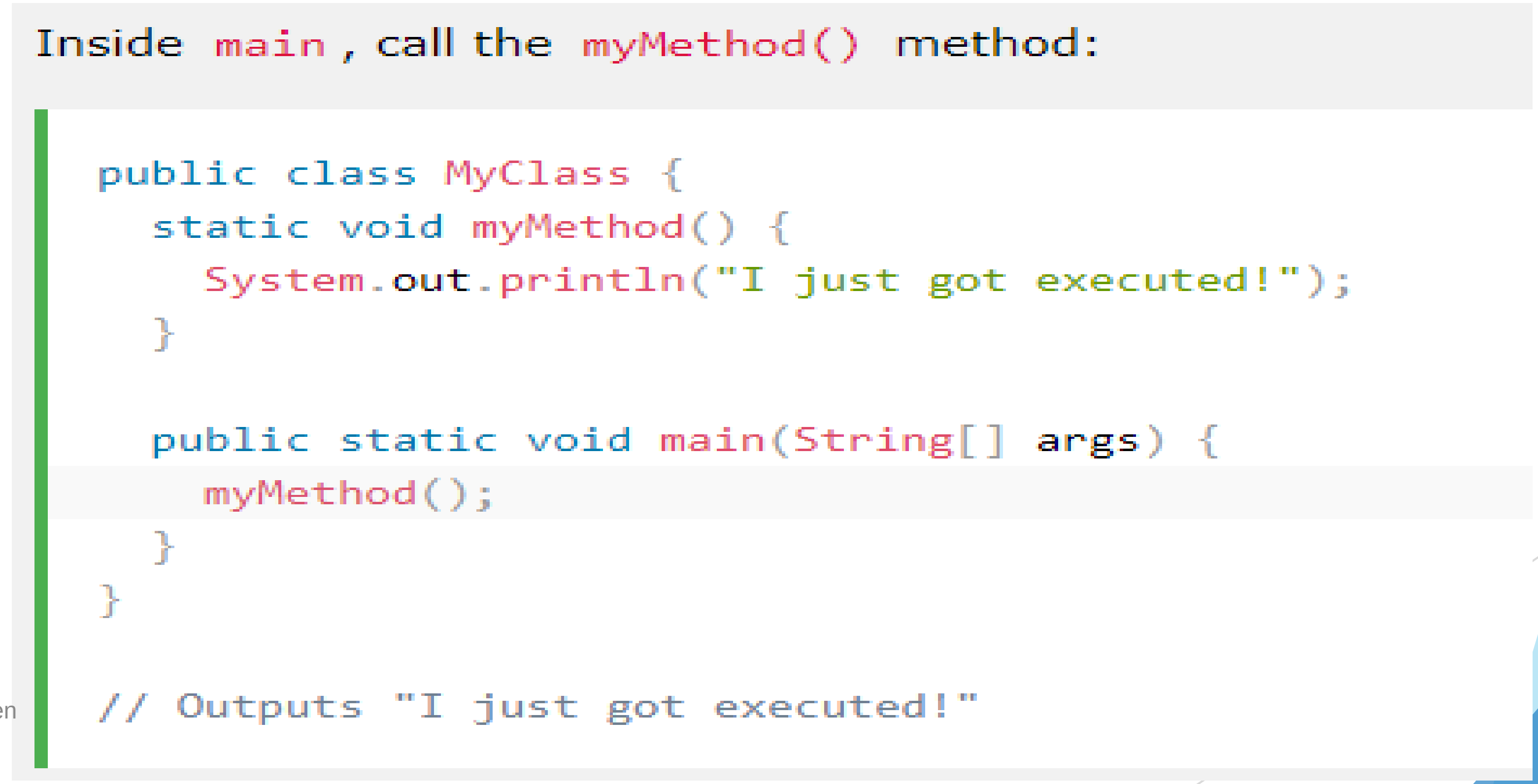
```
public class MyClass {
  static void myMethod() {
    // code to be executed
  }
}
```

Con…

- myMethod() is the name of the method.
- static means that the method belongs to the MyClass class and not an object of the MyClass class.
- void means that this method does not have a return value. You will learn more about return values later.

Call Method

- To call a method in Java, write the method's name followed by two parentheses () and a semicolon;
- In the following example, myMethod() is used to print a text (the action), when it is called:



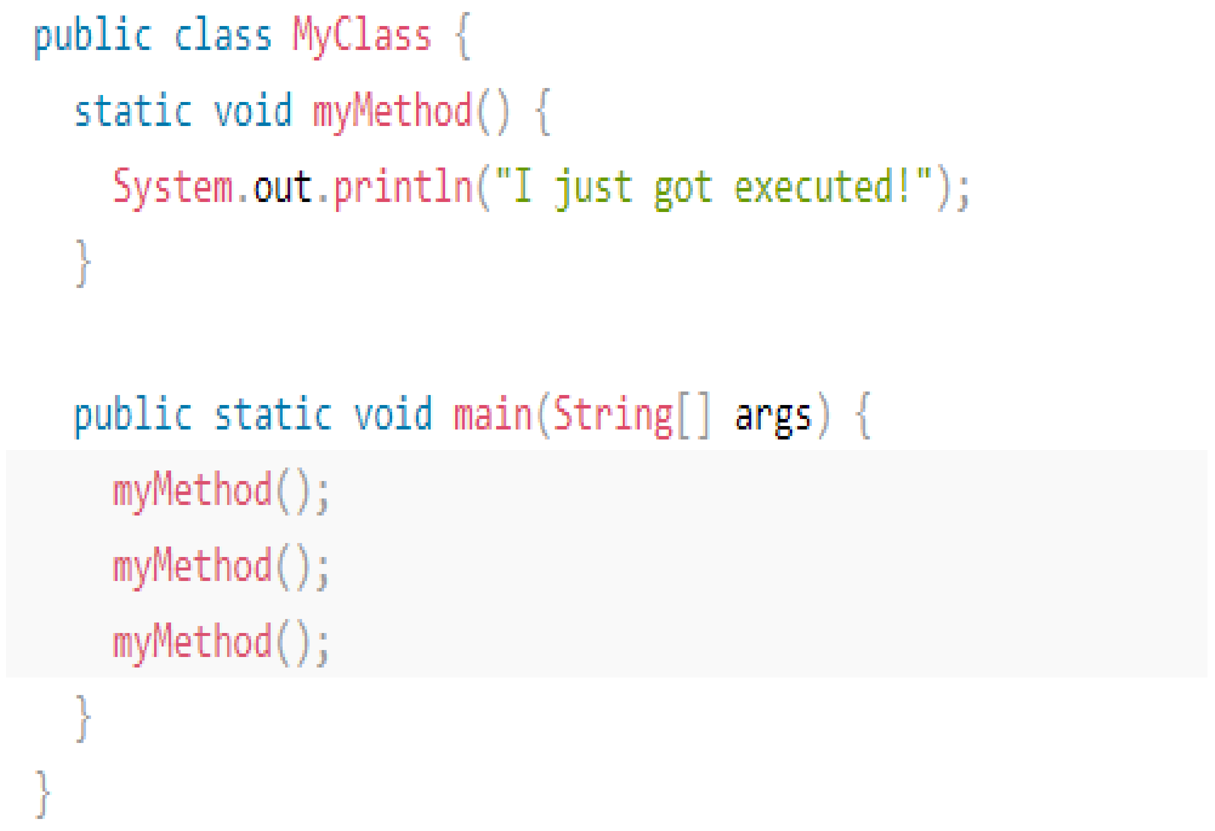
Inside main, call the myMethod() method:

```
public class MyClass {
  static void myMethod() {
    System.out.println("I just got executed!");
  }

  public static void main(String[] args) {
    myMethod();
  }
}

// Outputs "I just got executed!"
```

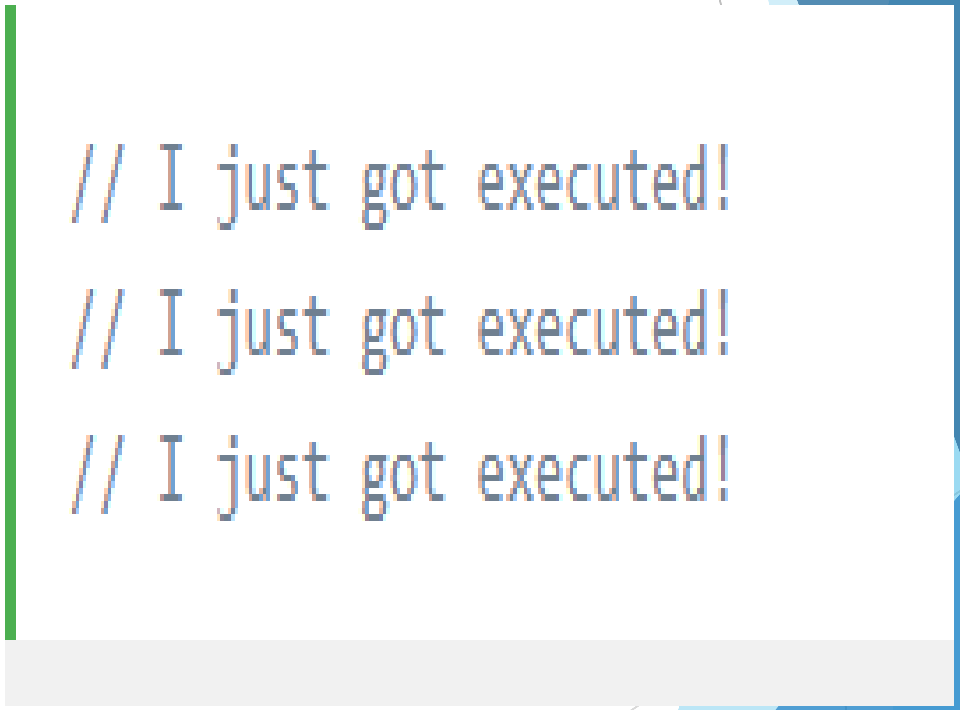
Calling of multiple Method



```
public class MyClass {
  static void myMethod() {
    System.out.println("I just got executed!");
  }

  public static void main(String[] args) {
    myMethod();
    myMethod();
    myMethod();
  }
}
```

Output



```
// I just got executed!
// I just got executed!
// I just got executed!
```

Practice

- **A vehicle's travel time can be calculated as:**

  Time=Distance/Speed.

- **Write a method getTravelTime that accepts the distance and speed as arguments and returns the vehicle's travel time.**

- **Demonstrate the method by calling it in a program that asks the user to enter values for distance and speed.**

```
import java.util.Scanner;
public class problem8 {
 public static void main (String [] args){
Scanner kb = new Scanner(System.in);
System.out.println("Please Enter the distance");
double distance = kb.nextDouble();
System.out.println("Please Enter the speed");
double speed = kb.nextDouble();
System.out.print(getTravelTime(distance, speed));
  }
public static double getTravelTime (double distance, double speed){
double time = distance/speed;
return time; } }
```

What is the output of the following program

```
class SquareMain {
public static void main(String[] args)
{
int result;
result = square();
System.out.println("Squared value of 
10 is: " + result);
}
public static int square() {
// return statement
return 10 * 10;
}
}
```

Aya Saleh Abd El_Mohsen

```
Squared value of 10 is: 100
```

Thanks