

Programming Fundamentals II

Lab 6: Methods

Learning Objectives:

- **Defining Methods with Formal Parameters:** Learn how to create methods that accept inputs (parameters) when they are called.
- **Invoking Methods with Actual Parameters (Arguments):** Understand how to call methods using specific values (arguments) for the parameters.
- **Defining Methods with a Return Value:** Discover how to create methods that perform calculations or operations and then return a result.
- **Defining Methods Without a Return Value:** Learn how to define **void** methods, which perform operations but do not return a result, and understand the differences between **void** methods and value-returning methods.
- **Passing Arguments by Value:** Understand the concept of passing arguments by value, where a copy of the argument's value is passed to the method.
- **Using Method Overloading and Understanding Ambiguous Overloading:** Explore how to create multiple methods with the same name but different parameters (overloading) and understand how to avoid ambiguous overloading, where the compiler cannot determine which method to call.
- **Determining the Scope of Variables:** Learn about the scope of variables, which defines where in the program a variable can be accessed or modified.

Requisite knowledge:

- Chapter 6 Elementary Programming from "Introduction to Java Programming Brief Version" Reference (See lms.uofk.edu).
- Lecture 6 (See lms.uofk.edu).

Lab 1 Assignment: Check Lab 6 Assignment file.

```

package tesmax;

public class TesMax {
    /** Main Method */
    public static void main(String[] args) {
        int i = 5;
        int j = 2;
        max(i,j); //invoke , call == statment

    }

    /** Return the max of two numbers */
    public static void max(double i ,int j ){

        double result ;
        if(i>j){ // true
            result = i; //5
        }
        else{
            result =j;
        }
        System.out.println("Max of "+ i +" and "+j+ " is " + result);
    }
}

```

```
package testvoidmethod;

public class TestVoidMethod {
    /** Main Method */
    public static void main(String[] args) {
        System.out.print("The grade is ");

        double score = 78.5;

        if(score >= 90.0){
            System.out.println('A');
        }
        else if(score >= 80.0){
            System.out.println('B');
        }
        else if(score >= 70.0){
            System.out.println('C');
        }
        else if(score >= 60.0){
            System.out.println('D');
        }
        else {
            System.out.println('F');
        }
    }
}
```

```

/** Passing Parameters */
package lab5;

public class Lab5 {
    /** Main Method */
    public static void main(String[] args) {
        String name = "Java Lab 6"; //5 times
        nPrintln("this is another string",8);
    }

    public static void nPrintln(String message, int n){
        for (int i=0; i < n; i++)
            System.out.println(message);
    }
}

```

```

package increment;
public class Increment {
    /** Main Method */
    public static void main(String[] args) {
        int x = 1;
        System.out.println("Before the call, x is "+x);
        increment(x);
        System.out.println("After the call, x is "+x);
    }

    public static void increment(int n){
        n++; // n=n+1; lab2 //2
        System.out.println("n inside the method is "+n);//2
    }
}

```

```

package testpassbyvalue;

public class TestPassByValue {
    /** Main Method */
    public static void main(String[] args) {
        // Declare and initialize variables
        int num1 = 1; //num2 = 2
        int num2 = 2; // num1 = 1

        System.out.println("Before invoking the swap method, num1 is "
            + num1 + " and num2 is " + num2);

        // Invoke the swap method to attempt to swap two variables
        swap(num1, num2); // false swap

        System.out.println("After invoking the swap method, num1 is " +
            num1 + " and num2 is " + num2);

    }

    /** Swap two variables */
    public static void swap(int n1, int n2) {
        System.out.println("\tInside the swap method");
        System.out.println("\t\tBefore swapping, n1 is " + n1
            + " and n2 is " + n2);

        // Swap n1 with n2
        int temp = n1;
        n1 = n2;
        n2 = temp;

        System.out.println("\t\tAfter swapping, n1 is " + n1
            + " and n2 is " + n2);
    }
}

```

Understanding Overloading function and Scope of variables:

```
public class Lab 6 {

    public static void main(String[] args) {

        // Test method overloading
        System.out.println("The maximum of 3 and 4 is " +
TestMethodOverloading.max(3, 4));
        System.out.println("The maximum of 3.0 and 5.4 is " +
TestMethodOverloading.max(3.0, 5.4));
        System.out.println("The maximum of 3.0, 5.4, and 10.14 is " +
TestMethodOverloading.max(3.0, 5.4, 10.14));

        TestMethodOverloading.method1();
    }
}

class TestMethodOverloading {

    public static int max(int num1, int num2) {
        if (num1 > num2)
            return num1;
        else
            return num2;
    }

    public static double max(double num1, double num2) {
        if (num1 > num2)
            return num1;
        else
            return num2;
    }

    /** Return the max of three double values */
    public static double max(double num1, double num2, double num3) {
        return max(max(num1, num2), num3);
    }

    /** Sample method with loops to understanding the Scope of variables */
    public static void method1() {
        int x = 1;
        int y = 1;
        for (int i = 1; i < 10; i++) {
```

```
        x += i;
    }
    for (int i = 1; i < 10; i++) {
        y += i;
    }

    // Print the values of x and y for verification
    System.out.println("The value of x after the loop is " + x);
    System.out.println("The value of y after the loop is " + y);
}
}
```

Output:

```
The maximum of 3 and 4 is 4
The maximum of 3.0 and 5.4 is 5.4
The maximum of 3.0, 5.4, and 10.14 is 10.14
```

Case Study: Generating Random Characters

```
package javaapplication44;

public class JavaApplication44 {

    public static void main(String[] args) {

        System.out.println("Math.random = " + Math.random()); // 0 < value < 1.0

        System.out.println("a = " + (int)'a');
        //
        System.out.println("z = " + (int)'z');
        //
        System.out.println("a - z = " + ('z' - 'a'));
        //
        System.out.println("a - z + 1 = " + ('z' - 'a' + 1));
        //
        System.out.println(" Math.random()*(a - z + 1) = " + ((Math.random()*('z' - 'a' + 1))));
        //
        int x = (int)('a' + Math.random()*('z' - 'a' + 1));

        System.out.print("'a' + Math.random()*('z' - 'a' + 1) = ");

        System.out.println(x);
        //
        System.out.printf("The random letter is : %c\n", ((char)x));

    }

}
```



```

package randomcharacter;
public class RandomCharacter {

    /** Generate a random character between ch1 and ch2 */
    public static char getRandomCharacter(char ch1, char ch2) {
        return (char)(ch1 + Math.random() * (ch2 - ch1 + 1)); //a-z , 0-26
    }
    /** Generate a random lowercase letter */
    public static char getRandomLowerCaseLetter() {
        return getRandomCharacter('a', 'z');
    }
    /** Generate a random uppercase letter */
    public static char getRandomUpperCaseLetter() {
        return getRandomCharacter('A', 'Z');
    }
    /** Generate a random digit character */
    public static char getRandomDigitCharacter() {
        return getRandomCharacter('0', '9');
    }
    /** Generate a random character */
    public static char getRandomCharacter() {
        return getRandomCharacter('\u0000', '\uFFFF');
    }

    public static void main(String[] args) {

        final int NUMBER_OF_CHARS = 175;
        final int CHARS_PER_LINE = 25;

        // Print random characters between 'a' and 'z', 25 chars per line
        for (int i = 0; i < NUMBER_OF_CHARS; i++) {
            char ch = getRandomLowerCaseLetter();
            if ((i + 1) % CHARS_PER_LINE == 0)
                System.out.println(ch);
            else
                System.out.print(ch);
        }
    }
}

```

End Of Lab!