

# Tutorial 5: Methods & Recursion

**Aim:**

- Get familiar with Java methods and recursion
- Consolidate learning from week 4, 5 and 6.
- Get feedback.

Note: Use the material from the lecture if you need. If you find any problem or have a question, ask your tutor. If you do not have enough time during the session, it is recommended that you finish the exercises at home. Challenges are optional, although recommended.

## **Section 01: Main Questions**

### **Q1: Hello With Methods**

Write a program that calls a method to print "Hello world".

Create a new project, and add the following method after your main method:

```
private static void hello() {  
    System.out.println("Hello world!");  
}
```

Add a call to your new method hello in your main:

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

You code should look like this:

```
public class Tutorial4_Ex3 {  
    public static void main(String[] args) {  
        hello();  
    }  
  
    private static void hello() {  
        System.out.println("Hello world!");  
    }  
}
```

## Q2: Calculator – Practice for Lab based Assessment

Write a program for a calculator that first displays the following menu: asks the user to select an option and reads the option. If the option selected is 0, the program should end; If the option selected is 1 or 2, the menu should be displayed again.

```
*****
```

```
    MENU
```

```
*****
```

```
1.- Addition
```

```
2.- Subtraction
```

```
0.- Quit
```

```
Please select an option:
```

When the user selects 1, the program should call a **method named “Addition”** to perform an addition by asking the user to introduce two double numbers. When the user selects 2, the program should call a **method named “Subtraction”** to perform a subtraction, by asking the user to introduce two double numbers. **After performing addition or subtraction, the program should display the menu again.**

## Q3: Printing Numbers

Write a program in Java that prints the numbers 1 to 100 without using a loop. Call the method from the main. You can use parameters if you need.

## Q4: Printing numbers using Recursion

- Write a **recursive method** that takes a positive number as an input parameter and prints all the numbers from that number to 0. For example, if you input the value 5 in the method should print:  
**5, 4, 3, 2, 1, 0.**
- In your main, write a program that asks the user to enter a positive number and calls your recursive method.

## Q5: Exponents using Recursion.

Write a program that computes exponents. The program should ask the user to input a number (int) and an exponent (int) and return the result. **You cannot use a loop, you have to use recursion (a method that calls itself).**

Example:

$$2^4 = 2 * 2 * 2 * 2 = 16.$$

$$6^3 = 6 * 6 * 6 = 216.$$

## Section 02: Challenging Questions

### Q5: Double recursion.

Let  $t(n)$  be a sequence defined for all non-negative integers by the following recursive definition:

$$t(0) = 1 \quad t(1) = 3 \quad t(n) = t(n-2) + t(n-1), \text{ for } n \geq 2$$

Write a program in Java that finds out the value of  $t(4)$  using recursion.

*Acknowledgement: This exercise was taken from Mathematics in Computing (4COSC002W).*

### Q6: Code verification

What does this display?

```
public static void main(String[] args) {  
    int aValue = 3;  
    multiply(aValue, 2);  
}  
private static void multiply(int sum1, int sum2){  
    sum1=sum1*sum2;  
    System.out.println(sum2+" "+sum1);  
}
```

- Rewrite it so the method returns the final value to the main program and then displays it.
- Can you make it return a double ex: return (double) sum1;

### Q7: Pass power of 2 of a number

Write a method which will take in a number and give back two times the number given. Write the main body of the program to call the function, and when a value is returned the main body should display it.

### Q8: Find factorial using Recursion

Write a program that will take a number and return the result of multiplying all the numbers together from 1 to the number given. (factorial). Write the main body of the program to call the function, and when a value is returned the main body should display it.

**Q9. Code verification II**

```
//swaps parameter names around in function call public
static void main(String[] args) {
    int a = 2;
    int b = 5;
    int c = 4;
    mixup(a, b, 3);
}

private static void mixup(int b, int c, int a ) {
    a = b + c - a;
    System.out.println(a + " " + c);
}
```

**Q10. Code verification III**

What does this display

```
public class Main { // demonstrate scope and functions
static int aValue = 5;
    public static void main(String[] args) {
        int aValue = 6;
        int bValue = 4;
        process(bValue);
        System.out.println(aValue);
    }
    private static void process(int aValue) { //AA
        aValue = aValue + 4;
        System.out.println(aValue);
    }
}
```

a. How different is the display if line AA above was:

```
private static void process(int cValue) { //AA
```

- b. Can you turn the procedure into a function that returns an int and then displays it.
- c. Can you turn the procedure into a function to return a double?

**Q11. Code verification IV**

Does this multiply 3\*2 or add 3 to 2? Why?

```
public static void main(String[] args) {  
    double Num1 = 3.0;  
    int Num2 = 2;  
    double total;  
    total = processA( Num1, Num2);  
    System.out.println(total);  
}  
  
private static double processA(double Sum, int Count) {  
    double newAnswer = Sum + (double) Count;  
    return newAnswer;  
}  
  
private static double processA(int Sum, double Count) {  
    double newAnswer = (double) Sum * Count;  
    return newAnswer;  
}
```

How would you make them both return back an 'int'?

**Q12. palindrome**

A nonnegative integer is called a palindrome if it reads forward and backward in the same way. For example, the numbers 5, 121, 3443, and 123454321 are palindromes. Write a method that takes as input a nonnegative integer and returns true if the number is a palindrome; otherwise, it returns false. Also write a program to test your method.

**Section 03 : HackerRank Challenges**

**HackerRank: Interview questions.** Solve the following tasks in HackerRank:

1. Java method overriding

**[Formative Assessment] ICT mock-up 1**

Practice for the ICT in Blackboard.