



## JOB SHEET 10

### Function 1

#### 1. Objective

- Students are able to understand the use of static functions in Java with parameters and return values
- Students are able to create programs using static functions and execute these functions

#### 2. Laboratory

##### 2.1 Experiment 1

1. Create a new project
2. Create a new class, name it **Greeting**
3. Create a function called **giveGreeting** inside the class

```
public class Greeting {  
  
    static void giveGreeting() {  
        System.out.println("Hello! Good morning");  
    }  
}
```

4. Create a **main** function inside the class, and execute the **giveGreeting** function from within the **main** function.



```
public class Greeting {  
  
    static void giveGreeting() {  
        System.out.println("Hello! Good morning");  
    }  
  
    public static void main(String[] args) {  
        giveGreeting();  
    }  
}
```

5. Compile and run the program

## 2.2Experiment 2

1. Using the class that was created in Experiment 1, add function called **saySomething** inside the **Greeting** class

```
public class Greeting {  
  
    static void giveGreeting() {  
        System.out.println("Hello! Good morning");  
    }  
  
    static void saySomething(String expression) {  
        System.out.println(expression);  
    }  
  
    public static void main(String[] args) {  
        giveGreeting();  
    }  
}
```



2. Execute the **saySomething** function from inside the **main** function

```
public class Greeting {  
  
    static void giveGreeting() {  
        System.out.println("Hello! Good morning");  
    }  
  
    static void saySomething(String expression) {  
        System.out.println(expression);  
    }  
  
    public static void main(String[] args) {  
        giveGreeting();  
        String exp = "Welcome to Java Programming";  
        saySomething(exp);  
    }  
}
```

3. Compile and run the program

## 2.3 Experiment 3

1. Create a new class, name it **Square**
2. Create a function named **squareArea** inside that class which returns the value **area** (int), with the input parameter **side** (int)



```
public class Square {  
  
    static int squareArea(int side) {  
        int area = side * side;  
        return area;  
    }  
}
```

3. Create a **main** function inside the class, and execute the **squareArea** function from within the **main** function.

```
public class Square {  
  
    static int squareArea(int side) {  
        int area = side * side;  
        return area;  
    }  
  
    public static void main(String[] args) {  
        int a = squareArea(5);  
        System.out.println("Area of a square with side = 5 is " + a);  
    }  
}
```

4. Compile and run the program

## 2.4 Experiment 4

1. Create a new class, name it **ArithmeticOperation**
2. Create a function named **multiplication** inside that class which returns the value **H** (int) and input parameters **C** and **D** (int)



```
public class ArithmeticOperation {  
  
    static int multiplication(int C, int D) {  
        int H;  
        H = (C + 10) % (D + 19);  
        return H;  
    }  
}
```

3. Create a function called **subtraction** inside that class which returns the value **X** (int) and input parameters **A** and **B** (int) and calls the **multiplication** function.

```
public class ArithmeticOperation {  
  
    static int multiplication(int C, int D) {  
        int H;  
        H = (C + 10) % (D + 19);  
        return H;  
    }  
  
    static int subtraction(int A, int B) {  
        int X;  
        A = A + 7;  
        B = B + 4;  
        X = multiplication(A, B);  
        return X;  
    }  
}
```

4. Create a **main** function inside the class, and execute the **subtraction** function from within the **main** function. Don't forget to add the Scanner library



```
public static void main(String[] args) {  
    int value1, value2;  
    Scanner input = new Scanner(System.in);  
    System.out.print("Input value 1: ");  
    value1 = input.nextInt();  
    System.out.print("Input value 2: ");  
    value2 = input.nextInt();  
    int result = subtraction(value1, value2);  
    System.out.println("The result is " + result);  
}
```

5. Compile and run the program.

## 2.5 Experiment 5

1. Create a new class, name it **MultiParameter**
2. Create a function called **Print** (void) inside the class using two types of parameter data, namely String and int

```
public class MultiParameter {  
  
    static void Print(String str, int... a) {  
        System.out.println("String: " + str);  
        System.out.println("Number of parameters: " + a.length);  
        for (int i : a) {  
            System.out.print(i + " ");  
        }  
        System.out.println("");  
    }  
}
```

3. Create a **main** function inside the class, and execute the **Print** function from



within the **main** function.

```
public static void main(String[] args) {  
    Print("Basic Programming", 85, 90);  
    Print("Information Technology", 1, 2, 3, 4, 5);  
    Print("Politeknik Negeri Malang");  
}
```

6. Compile and run the program.

## 2.6 Experiment 6

1. Create a new class, name it **Geometry1**
2. Create a program to calculate the area of a rectangle and volume of blocks without using functions

```
public static void main(String[] args) {  
    Scanner input = new Scanner(System.in);  
    int length, width, height, area, volume;  
    System.out.print("Enter a length value: ");  
    length = input.nextInt();  
    System.out.print("Enter a width value: ");  
    width = input.nextInt();  
    System.out.print("Enter a height value: ");  
    height = input.nextInt();  
    area = length * width;  
    System.out.println("Area of rectangle is " + area);  
    volume = length * width * height;  
    System.out.println("Volume of block is " + volume);  
}
```

3. Create another new class, name it **Geometry2**
4. **Geometry2** contains the program code for calculating the area of a rectangle and the volume of a block by using a function, so that there are three functions, namely **calculateArea**, **calculateVolume**, and the **main** function

- calculateArea function

```
static int calculateArea(int lgt, int wdt) {  
    int a = lgt * wdt;  
    return a;  
}
```

- calculateVolume function

```
static int calculateVolume(int hgt, int a, int b) {  
    int vol = calculateArea(a, b) * hgt;  
    return vol;  
}
```

- main function

```
public static void main(String[] args) {  
    Scanner input = new Scanner(System.in);  
    int length, width, height, area, volume;  
    System.out.print("Enter a length value: ");  
    length = input.nextInt();  
    System.out.print("Enter a width value: ");  
    width = input.nextInt();  
    System.out.print("Enter a height value: ");  
    height = input.nextInt();  
    area = calculateArea(length, width);  
    System.out.println("Area of rectangle is " + area);  
    volume = calculateVolume(height, length, width);  
    System.out.println("Volume of block is " + volume);  
}
```

7. Compile and run the two programs (class Geometry1 and Geometry2)
8. Describe the flow of the program for calculating the area of a rectangle and volume of blocks in class Geometry2





## Questions!

1. Based on Experiments 2 and 3, explain when a function requires a return value!
2. In Experiment 4, add a function that is used to ensure that the **value1** and **value2** are at least 0, then call that function in the **main**!
3. Explain why the parameter entries in Experiment 5 are written with `int ... a`!
4. What is the output of the program below, then explain the flow of the program!

```
public class MyProgram {  
  
    static void printUntil(int i) {  
        for (int j = 1; j <= i; j++) {  
            System.out.print(j);  
        }  
    }  
  
    static int total(int num1, int num2) {  
        return (num1 + num2);  
    }  
  
    static void printTotal(int num1, int num2) {  
        printUntil(total(num1, num2));  
    }  
  
    public static void main(String[] args) {  
        int temp = total(1, 1);  
        printTotal(temp, 5);  
    }  
}
```

## 3. Assignment

1. Create a static method called **Max3** (`int bil1, int bil2, int bil3`) which takes three integer parameters and returns an integer number which is the maximum value among the three numbers. Note: You can create other static



methods besides **Max3**. After that, call the **Max3** static method in your **main** method.

2. Create a class called **Circle** in which there is a function to calculate the circumference of a circle and the area of a circle.
3. Create a program to fill array **B** with the data type int (10 students' test scores), where the input and filling process into the array is carried out in a function. Next, create another function to calculate the average value of the array (the average score of student tests). Print the average value, with the instructions for printing in the **main** function.