

Methods II

```
public static int FunctionName(String a, double b) {  
    // code  
}
```

public is the access modifier. You will learn more about this later.

static means that the method belongs to the class and not an instance of the class.

int is the return type. Possible return types are **int**, **double**, **String**, **boolean**, **void**, etc. **FunctionName** is the name of the function.

String a, **double b** are the parameters. You can have as many parameters as you want, but they must be separated by commas. Parameters are variables that are passed into the function. You can use them in the function as if they were declared in the function.

- Parameter names and argument names can be different.

```
public static void main(String[] args) {  
    int a = 5;  
    int b = 4;  
  
    // a, b passed as argument to sum function  
    sum(a, b);  
}  
  
// parameter names are x, y  
public static int sum(int x, int y){  
    return x+y;  
}
```

- In Java, functions can be overloaded. This means that you can have multiple functions with the same name, but different parameters.

```
public static void main(String[] args) {  
    int a = 5;  
    int b = 4;  
    int c = 1;  
  
    sum(a, b); // calls A  
    sum(a, b, c); // calls B  
}  
  
// A  
public static int sum(int x, int y){  
    return x+y;  
}  
  
// B  
public static int sum(int x, int y, int z){  
    return x+y+z;  
}
```

Fun extra fact: `println()` is a method that was overloaded in the class `System`. You can write:

```
System.out.println("same"); //String
System.out.println(24); //Integer
System.out.println(24.6); //Double
System.out.println(a==b); //Boolean
```

- In Java, primitive types (int, char, double, float etc.) are passed-by-value. Objects are passed-by-reference. (you will learn about pass-by-reference later)

```
public static void main(String[] args) {
    int a = 5;
    int b = 4;

    // cannot destroy the variables a, b
    destroyer(a, b);

    System.out.println(a); // 5
    System.out.println(b); // 4
}

public static void destroyer(int a, int b){
    // we are changing the local copies of a, b
    a = 0;
    b = 0;
}
```

Exercises

1. Which statement should we put in the loop to complete the function which is supposed to return the reverse of its input?

```
public class punchcard {
    public static String reverse(String s) {
        String res = "";
        for(int i=0; i<s.length(); i++) {
            -----
        }
        return res;
    }

    public static void main(String[] args) {
        String s = "Hello World!";
        System.out.println(reverse(s));
    }
}
```

- a) `res += s.charAt(i);`
- b) `res = s.charAt(i) + res;`
- c) `s.charAt(i) = s.charAt(s.length()-i);`
- d) `s.charAt(i) = s.charAt(s.length()-i-1);`

answer: b - `res = s.charAt(i) + res;`

2. What will be printed to screen when we run this program?

```
public class punchcard {
    public static void swap(int a, int b) {
        int temp = a;
        a = b;
        b = temp;
    }

    public static void main(String[] args) {
        int a = 5;
        int b = 2;
        swap(a, b);
        System.out.println(a);
    }
}
```

- a) 0
- b) 2
- c) 5
- d) Null
- e) error

answer: c - 5

3. What is wrong with the following function?

```
public class punchcard {
    public static String concat(String s1, String s2) {
        // concatenates two strings
        System.out.println(s1 + s2);
    }

    public static void main(String[] args) {
        // main method
        concat("Hello", "World");
    }
}
```

```
}  
}
```

- a) Comments can't be inside functions.
- b) static functions can't take parameters.
- c) Return type is String, but function does not return anything.
- d) + operator does not concatenate strings.

answer: c - Return type is String, but function does not return anything.

4. What is wrong with the following program?

```
public class punchcard {  
    public static int min(int a, int b) {  
        if(a < b)  
            return a;  
        else  
            return b;  
    }  
    public static void main(String[] args) {  
        int a = 3;  
        int b = 2;  
        int c = 5;  
        a = min(a,b,c);  
    }  
}
```

- a) If-else structure is missing curly braces.
 - b) Parameters of `min()` conflicts with variables `a`, `b` defined in main.
 - c) Variable `a` appears on both sides of assignment operator which causes undefined behavior.
 - d) The function `min` must be given two arguments, not three.
-

5. What is wrong with the following program?

```
public class punchcard {  
    public static innocentFunction() {  
        System.out.println("I am innocent.");  
    }  
    public static void main(String[] args) {  
        innocentFunction();  
    }  
}
```

- a) There cannot be two static functions within the same scope.
- b) A function must have a return type.
- c) There has to be more than one statements inside a function body.
- d) A function must return something.

answer: b - A function must have a return type.

6. In Java, a method name cannot start with a ...

- a) Number
- b) #
- c) - (hyphen)
- d) All the above

answer: d - All the above

7. What does the following code do?

```
public class punchcard {
    public static int mystery(int a, int b) {
        if(b==0)
            return a;
        else
            return mystery(a, b-1) + 1;
    }
    public static void main(String[] args) {
        System.out.println(mystery(2, 25));
    }
}
```

- a) Subtracts two numbers using recursive subtraction.
- b) Multiplies two numbers through a series of iterations.
- c) Adds two numbers in a recursive and inefficient manner.
- d) Divides two numbers by employing iterative division.

answer: Adds two numbers in a very inefficient way.

8. I want my function to calculate multiplication of positive integers. What should I put in the blank?

```
public class punchcard {
    public static int mult(int a, int b) {
        if(a == 1)
            -----
        else
            return mult(a-1, b) + b;
    }

    public static void main(String[] args) {
        System.out.println(mult(3, 5));
    }
}
```

- a) return 1;
- b) return 0;
- c) return a;
- d) return b;

answer: d - return b;

9. What will happen when we run the following program?

```
public class punchcard {
    public static void a() {
        b();
    }
    public static void b() {
        a();
    }

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

- a) The program will enter an endless loop without printing any output. (StackOverflowError)
- b) The program will execute successfully and terminate after printing an output.

- c) The program will encounter a compilation error due to recursive method calls.
- d) The program will not encounter any error, but it will not run either.

answer: a - The program will enter an endless loop without printing any output. (StackOverflowError)

10. What will happen when we call a(n) with a non-negative n value?

```
public class punchcard {
    public static void a(int n) {
        if(n==0)
            System.out.println("ends in a()");
        else
            b(n-1);
    }
    public static void b(int n) {
        if(n==0)
            System.out.println("ends in b()");
        else
            a(n-1);
    }

    public static void main(String[] Args) {
        a(5);
    }
}
```

- a) ends in a()
- b) ends in b()
- c) compilation error
- d) runtime error

answer: b - ends in b()

It ends in a() if n is even, and ends in b() if n is odd. Can you see why?

Let's say n=5

a(5) --> b(4) --> a(3) --> b(2) --> a(1) --> b(0) prints "ends in b()"

11. What is the output of following code?

```
public class punchcard {
    public static void Function(int a) {
        System.out.print("The input is integer " + a);
    }
    public static void Function(double a) {
        System.out.print("The input is double " + a);
    }

    public static void main(String[] Args) {
        Function(2);
    }
}
```

- a) The input is integer 2
- b) The input is double 2.0
- c) The input is integer 2 The input is double 2.0
- d) error

answer: a - The input is integer 2

12. In the last code what happens if we remove first function? What would be the output?

```
public class punchcard {  
    public static void Function(double a) {  
        System.out.println("The input is double " + a);  
    }  
  
    public static void main(String[] Args) {  
        Function(2);  
    }  
}
```

- a) The input is integer 2
- b) The input is double 2.0
- c) The input is integer 2 The input is double 2.0
- d) error

answer: b - The input is double 2.0