# **Methods:**

Methods allow us to reuse the code. For example, if you are adding two random numbers a lot of times, instead of writing it every time, you can write a method to add those number and call that method whenever you need to add two numbers.

**Defining a method:**

To define a method, you need to specify

Modifier – Defines access type of method

Return type – The data type of the value returned by the method

Method Name – The name of the method

Parameters list – Comma separated list of the input parameters.

Method body – The code you need to execute. It is enclosed in braces.

Example:

1. You can define a method as

public int add(int a, int b){

}

Here public is the modifier. Int is the return type. Add is the method name. int a and int b are parameters.

1. So, in the above method we are returning an integer.

If we return nothing from the method, we can define it as

public void add(int a, int b){

}

In the above, we don’t want to return anything so return type is void

1. Returning string

public String somemethod(String str1, String str2){

}

Here we are also passing strings.

1. Returning an array of integers

public int[] somemethod(int[] arr){

}

In the above method, we are returning an array of integers so mentioned as int[]

1. Returning an array of Strings

public String[] somemethod(String[]){

}

In the above method, we are returning an array of strings so mentioned as int[] and we are also passing an array of strings.

1. Passing and Returning an object

public obj change(obj o){

}

In the above method, if the class name of the method is obj, we have mentioned the return type as class name which means we are return an object of that class. Similarly in the parameters also we are passing an object.

**Calling a method:**

When you are calling a method, you need to capture the value returned by that method.

Example:

1. Calling a method that takes integers as inputs and returns an integer.

Method:

public int add(int a, int b){

}

Calling the method:

int a = add(2, 3);

Example:

**package** package1;

**public** **class** AddingNumbers {

**public** **int** add(**int** a, **int** b) {

**int** c = a+b;

**return** c;

}

**public** **static** **void** main(String[] args) {

**int** a = 2;

**int** b = 2;

AddingNumbers an = **new** AddingNumbers();

System.***out***.println(an.add(a,b));

}

}

1. Calling a method that takes integers as arguments and returns nothing.

Method:

public void add(int a, int b){

}

Calling the method:

add(2, 3)

Example:

**package** package1;

**public** **class** AddingNumbers {

**public** **void** add(**int** a, **int** b) {

**int** c = a+b;

System.***out***.println(c);

}

**public** **static** **void** main(String[] args) {

**int** a = 2;

**int** b = 2;

AddingNumbers an = **new** AddingNumbers();

an.add(a,b);

}

}

1. Calling a method that returns a string by passing strings

Method:

public String somemethod(String str1, String str2){

}

Calling the method:

String str3 = somemethod(“Subbu”, “Selenium”)

Example:

**package** package1;

**public** **class** StringMethodExamples {

**public** String concatenate(String str1, String str2) {

String str3 = str1+" "+str2;

**return** str3;

}

**public** **static** **void** main(String[] args) {

String str1 = "Subbu";

String str2 = "Selenium";

StringMethodExamples sme = **new** StringMethodExamples();

String str3 = sme.concatenate(str1, str2);

System.***out***.println(str3);

}

}

1. Calling a method that returns an array by passing an array

public int[] somemethod(int[] arr){

}

Calling the method:

int[] arr1 = {1, 2, 3};

int[] arr2 = somemethod(arr1)

Example:

**package** package1;

**public** **class** ExceptionExample {

**public** **static** **int**[] mth(**int**[] arr) {

**int**[] arr1 = arr;

**for** (**int** i=0; i<arr.length; i++) {

arr1[i] = arr[i]\*2;

}

**return** arr1;

}

**public** **static** **void** main(String[] args) {

**int**[] arr1 = {3,4,5};

**int**[] arr2 = *mth*(arr1);

System.***out***.println(arr2[0]);

System.***out***.println(arr2[1]);

System.***out***.println(arr2[2]);

}

}

1. Returning an array of Strings

public String[] concatenate(String[] arr){

}

Calling method:

String[] arr1 = {“Subbu”, “Selenium”};

String[] arr2 = concatenate(arr1);

**package** package1;

**public** **class** StringMethodExamples {

**public** String[] concatenate(String[] str) {

String[] str1 = str;

str1[0] = str[0]+" tutorials";

str1[1] = str[1]+" tutorials";

**return** str1;

}

**public** **static** **void** main(String[] args) {

String[] str = {"Subbu", "Selenium"};

StringMethodExamples sme = **new** StringMethodExamples();

String[] str1 = sme.concatenate(str);

**for**(**int** i=0; i<str1.length; i++) {

System.***out***.println(str1[i]);

}

}

}

1. Passing and Returning an object

public obj change(obj o){

}

Calling Method:

obj o1 = new obj();

obj o2 = change(o1);

Example:

**package** package1;

**public** **class** OPCA {

**public** **int** a;

**public** OPCA setVariable(OPCA opc) {

OPCA opc1 = **new** OPCA();

opc1.a = opc.a \* 10;

**return** opc1;

}

**public** **static** **void** main(String[] args) {

OPCA opc1 = **new** OPCA();

opc1.a = 2;

OPCA opc2 = **new** OPCA();

opc2 = opc2.setVariable(opc1);

System.***out***.println(opc2.a);

}

}