Methods

What is a method ?

A method is a block of statements to do some task. You can call the method from any place in the program. In other words it's a subprogram has input , process and output.

Each method must have the following:

return type **method name** (**parameters**)
{

}

Return type : Must be one of the primitives data types, class name, array, void.

Method name : Rules of method name is the same as identifier naming rules.

(Parameters): You can put as many as parameters as you need, be careful with the following:

- you must put parentheses even if you do not have parameters.
- You must define each variable alone for example suppose you have 3 integers :
  (int a,int b, int c) ✔
  (int a,b,c) ✘
  (int a; int b; int c) ✘
- (int a = 5) ✘

- if you have no parameters you put ().

You can add some modifiers to the method in the beginning of the method signature, for example :

```
public void display()
{

}
```

```
public static main(String aa[])
{
}
```

you can swap the position of the modifiers in the method signature, but you should put all of them in the beginning of the signature.

```
final static public void display()
{
    System.out.println("OK");
}

static final public void display()
{
    System.out.println("OK");
}
```

You should make the method do one task only and give it a name to reflect it's task.

You must have block {} for the method even if there is one statement.

The method returns zero value (void) or one value only.

The method will not change the values of the parameters. This is called ***call by value,*** when sending objects, it will use ***call by reference.***

Method examples:

```
public int max(int a, int b)
{
     int c;

     if (a > b)
          c = a;
     else
          c = b;

     return c;
}

public int max(int a, int b)
{
     if (a > b)
          return a;
     else
          return b;
}
```

must return int, or you get an error

When return is executing it will leave the method and return a value.

Can I put return in void method?

```
public void method1(int x)
{
     if (x < 0)
          return;

     System.out.println(x + " is positive");
}

public String result(double mark)
{
     if (mark >= 60)
          return "Passed";
     else
          return "Failed";
}
```

only return no value, it's void

How to call a method ?

Suppose we have a method with the following signature:

public int **max(int a, int b)** ***the red stuff is the method signature***

to call any method you should know the name, parameters required and return type, to call the previous method you do the following:

int x = max(10,3);

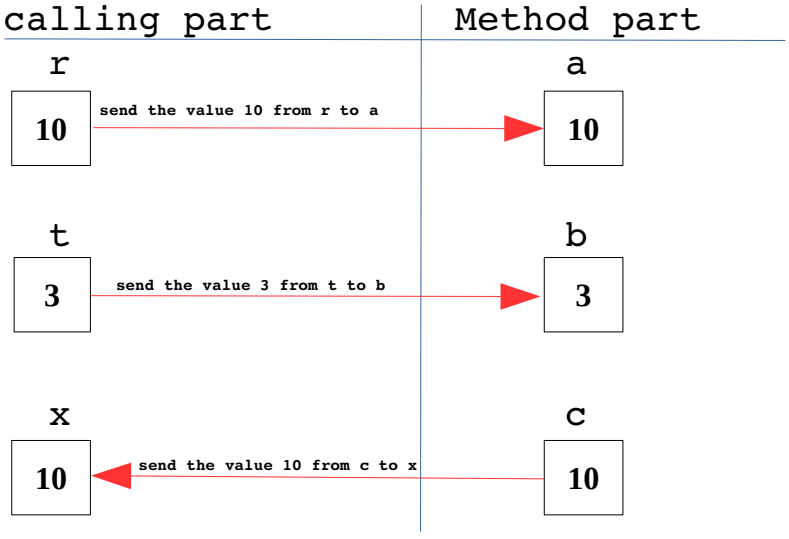
The variables or values between parentheses in calling statement called ***arguments***.

or you can call this way:

```
int r = 10;
int t = 3;
int x = max(r,t);
```

Inside the Memory

```
int x = max(r, t)
```



```
int x = max(r+3, t-2);
```
✔

```
int x = max(t = r+3,t+12);
```
???? ✔

```
int x = max(int r = 3,int t);
```
✘

In the arguments list you can not define any variable, when sending an argument you should have defined and given a value to this variable.

```
if (max(10,2) > 5)
```
✔

```
System.out.println(max(10,2));
```
✔

```
max(10,2);
```
✔
This is correct but you did not use the return value !

```
int x = max(10,2) + 33 + r;
```
✔

```
int x = max(max(10,2), 3);
```
✔

If you have a method with **void** return value, you can call the method **only in one way**:

```
display();
```

You can not call it using if, println, while, etc.

Methods overloading

You can have more than one method with same name but with different set of parameters.

For example:

1- public int m1(int a,int b) → name is m1, parameters int , int

2= public int m1(int a) → name is m1, parameters int

3- public int m1(double a,int b) → name is m1, parameters double , int

4- public int m1(int a,double b) → name is m1, parameters int , double

5- public int m1(double a,double b) → name is m1, parameters double, double

if you call m1(1,3), which method will be executed ?

Method number 1, you are sending two int(s) to m1.

if you call m1(1.0,3), which method will be executed ?

Method number 3, you are sending double ,int to m1.

If we have public int m1(float a), and called it by m1(3.0), what will happen ?

You will get error! Why ?

Because 3.0 is double, you can not promote double to float.

To call this method do the following : *m1((float) 3.0)*

Promote to →	double	float	long	int	char	short	byte	boolean
double	✘	✘	✘	✘	✘	✘	✘	✘
float	✔	✘	✘	✘	✘	✘	✘	✘
long	✔	✔	✘	✘	✘	✘	✘	✘
int	✔	✔	✔	✘	✘	✘	✘	✘
char	✔	✔	✔	✔	✘	✘	✘	✘
short	✔	✔	✔	✔	✘	✘	✘	✘
byte	✔	✔	✔	✔	✘	✔	✘	✘
boolean	✘	✘	✘	✘	✘	✘	✘	✘

Note:

return type is not part of method signature.

Methods already defined in java:

There are many packages already defined in java, you can use them in your program, for example

java.lang

java.util

xjava.swing

these packages contains many classes we can use, in ***java.util*** we used Scanner class, each class has many methods, for example Scanner has nextInt(), nextDouble() and etc.

We have very helpful class in math operation, this class called Math (***lava.lang.Math***), some methods in this class:

Math.max(int,int)

Math.abs(x)

Math.sqrt(double)

You do not create an object from this class like Scanner, you can use the methods with the class name, Why?

Because these methods are defined as static method inside the Math class, when you have a static method in a class you call it by the class name.