Methods

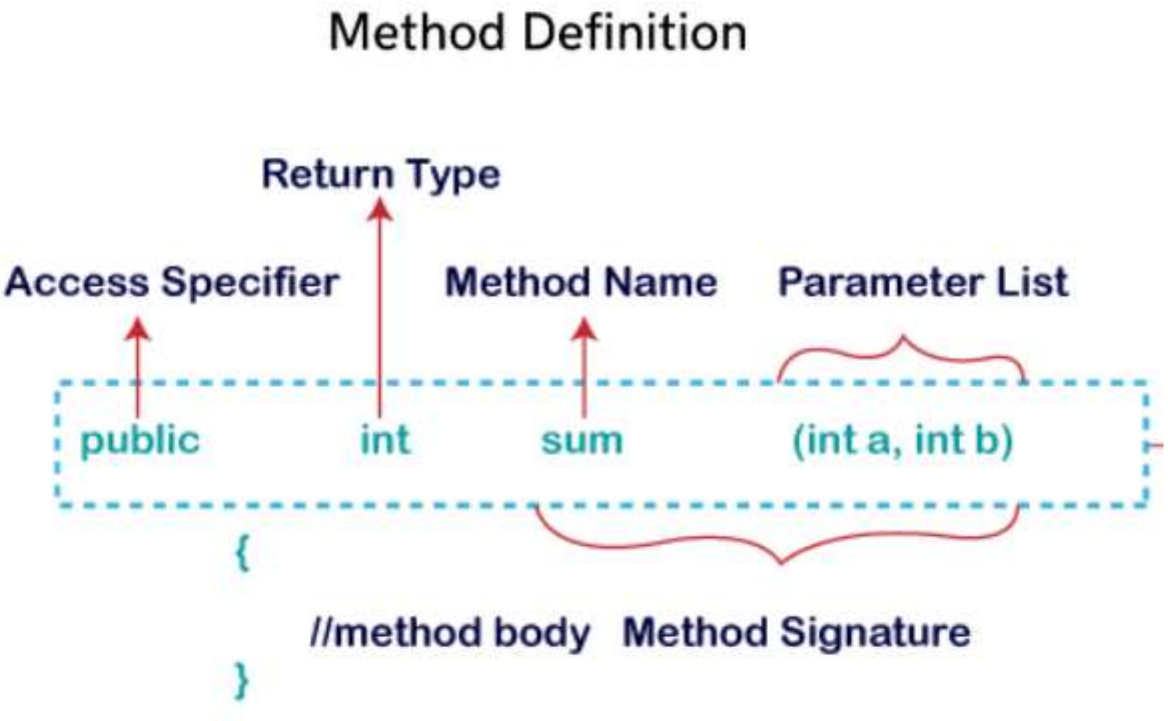
- Method is a block of instructions which is used to perform a specific task.
- **Syntax:**

```
[ access modifier] [modifier] return type name([datatype var1, datatype var2, …])
{


}
```

Terminologies:

- **Method signature**
  - Method name
  - Formal argument
- **Method declaration**
  - Access modifier
  - Modifier
  - Return type
  - Signature
- **Method definition**
  - Method declaration
  - Method body / implementation / block

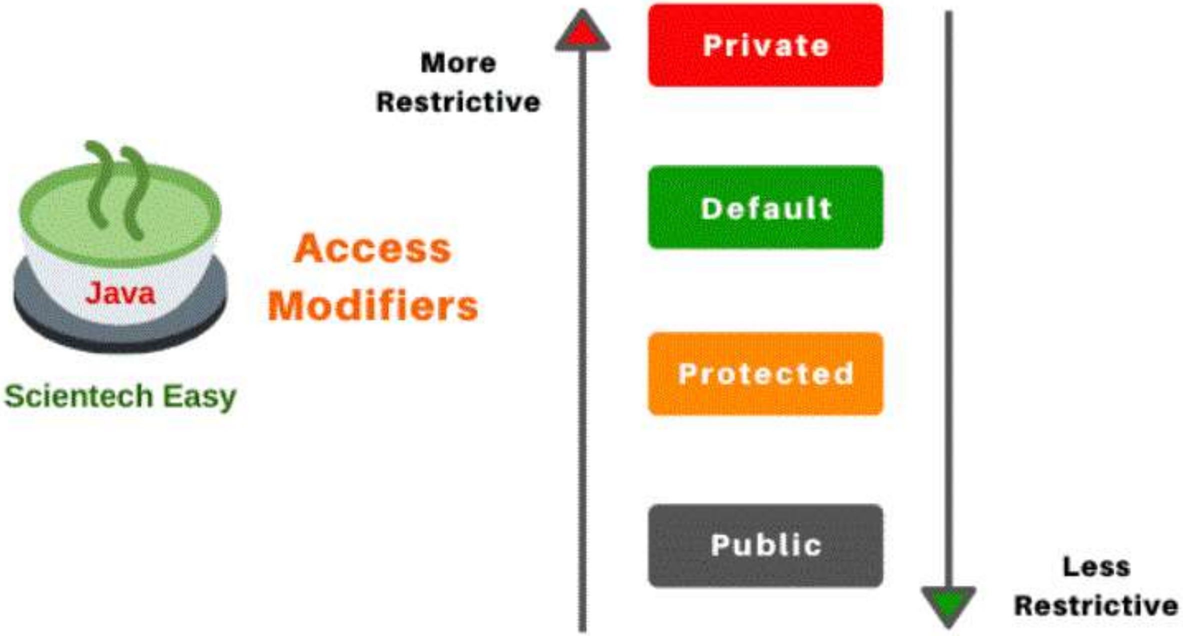


Modifiers: Modifiers are the keywords which are responsible to modify the characteristics of the members.

Example: static, final.

Access Modifiers

- Access modifiers are used to change the accessibility of a member. **We have four levels of access modifiers**
  - private
  - default
  - protected
  - public



RETURN TYPE

- The method after execution can return a value back to the caller.
- Therefore it is mandatory to specify what type of data is returned by the method in the method declaration statement, This is done with the help of return type.

Return type definition

- The return type is a data type that specifies what type of data is returned by the method after execution.

A method can have the following return types

- void
- primitive data type
- non-primitive data type

void

- void is a data type that is used as a return type when the method returns nothing.
- It is a keyword In java.

NOTE:

- A method can’t create inside another method.
- A class can have any number of methods.

NOTE :

- A method will get executed only when it is called, we can call a method with the help of a method call statement.

METHOD CALL STATEMENT :

- The statement which is used to call a method is known as a method call statement.

Syntax to create a method call statement :

methodName([Actual arguments]);

We can call a no-argument method without passing an actual argument in the method call statement.

METHOD CALL STATEMENT FLOW :

- Execution of calling method is paused
- Control is transferred to the called method.
- Execution of called method begins.
- Once the execution of the called method is completed the control is transferred back to the calling method.
- Execution of calling method resumes.

CALLING METHOD :

The method which is trying to call another method is known as the calling method (caller).

CALLED METHOD :

The method which is being called by the caller is known as a called method.

MAIN METHOD :

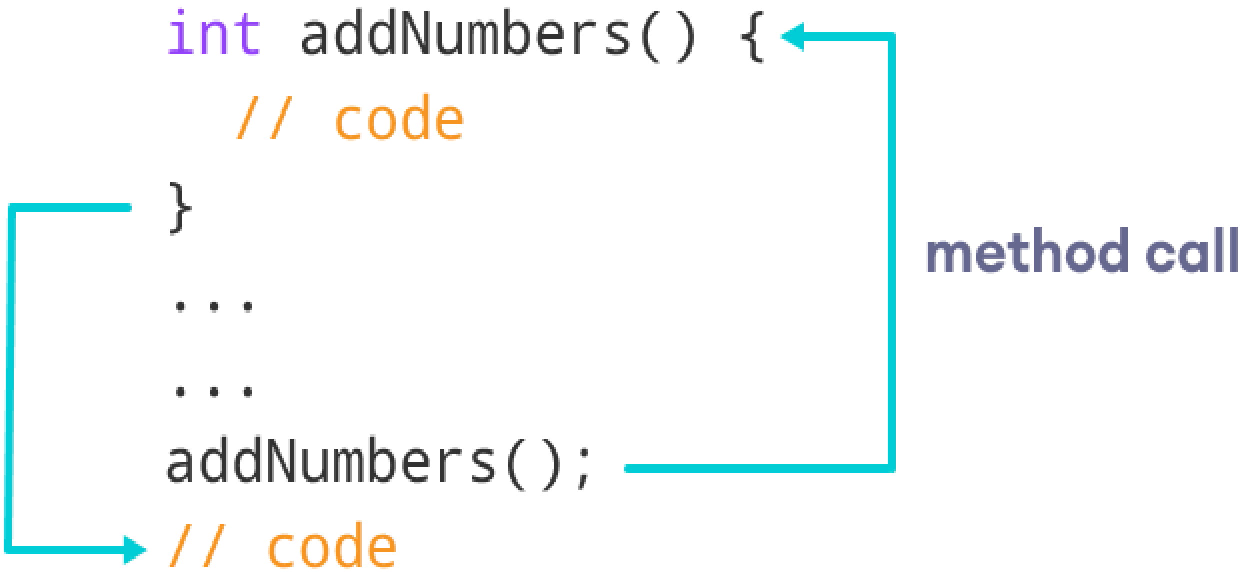
The execution of a java program always starts from the main method defined as follows

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

}
```

PURPOSE OF THE MAIN METHOD :

- Start the execution
- Control the flow of the execution
- End of execution



```
int addNumbers() {
  // code
}
...
...
addNumbers();
// code
```

NOTE :

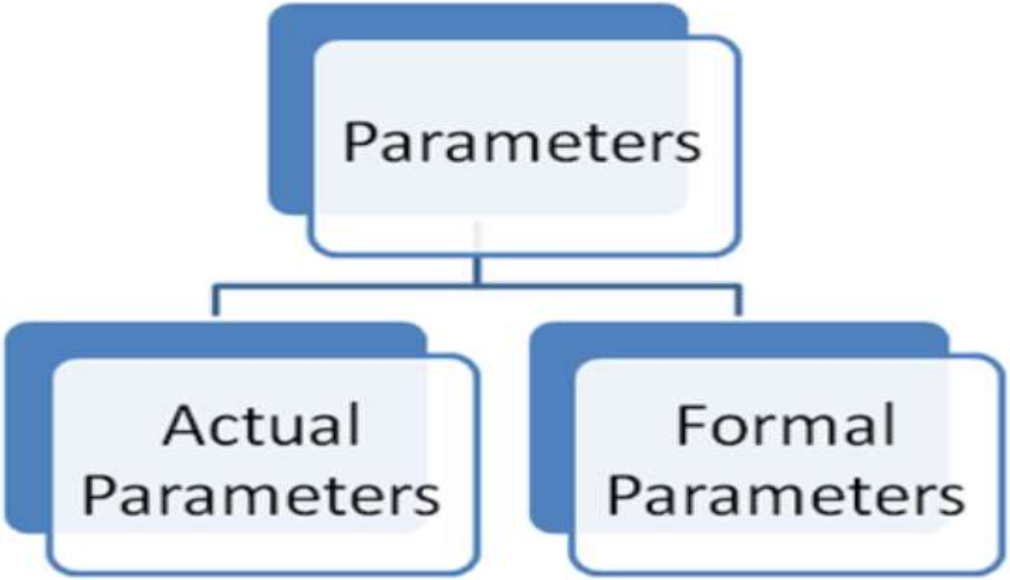
- A method can be executed only when it is called, we can call a method any number of times, therefore it is said to be code reusability.
- The main method is always called by JVM.

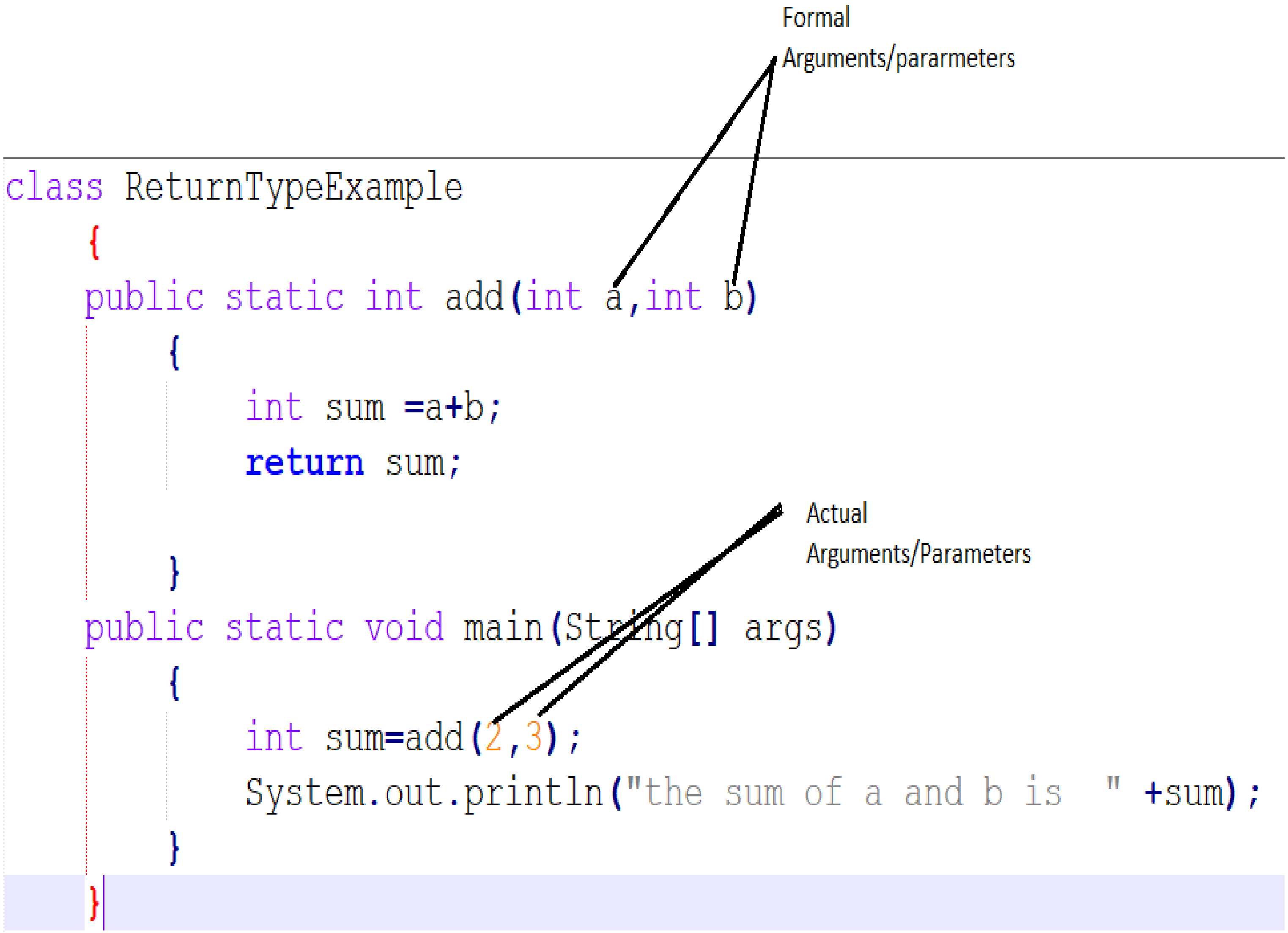
- **Parameterized Method:** The method which has formal arguments is called as Parameterized method.

- Parameterized Methods are used to accept the data.

Formal arguments: The variable which is declared is known as Formal argument.

Actual arguments: The values passed in the method call statement is called Actual arguments.





Formal
Arguments/pararmeters

```
class ReturnTypeExample
    {
    public static int add(int a,int b)
        {
            int sum =a+b;
            return sum;

        }
    public static void main(String[] args)
        {
            int sum=add(2,3);
            System.out.println("the sum of a and b is  " +sum);
        }
    }
```

Actual
Arguments/Parameters

Rules to call the parameterized method

- The number of actual arguments should be same as the number of formal arguments.
- The type of corresponding actual arguments should be same as the type of formal arguments, if not complier tries implicit conversion if it is not possible then we get compile time error.

return Statement:

- return is a keyword.
- It is a control transfer statement.
- When the return statement is executed, the execution of the method is terminated and control is transferred to the calling method.

Steps to use return statement

- **Step1:**

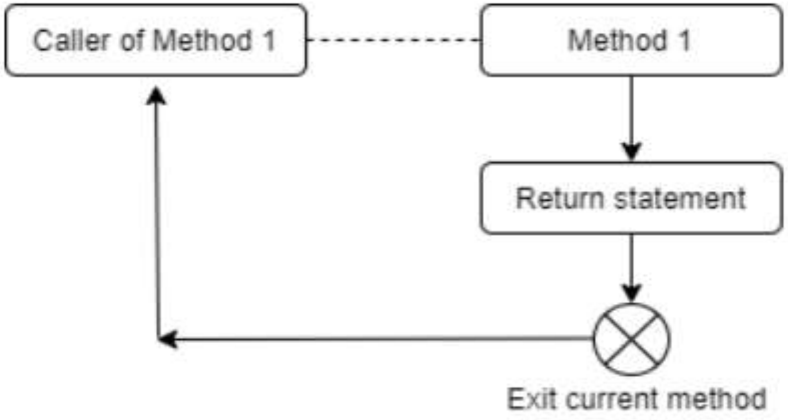
➢ Provide a return type for a method(it should not be void)

- **Step2:**

➢ Use return statement in the value to be returned.

Rule:

- The type specified as return type should be same as the type of value passed in the return statement.



Flow Chart of return statement