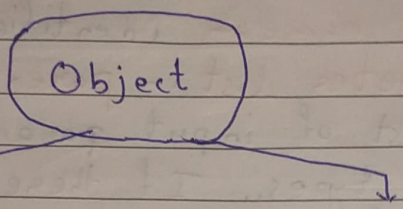
Page No:
Date:

Methods in Java.



Has-part
State / Properties
[Fields / Variables]

Does-part.
Behavior / Activity
[Methods / Functions]

① A method, like a function, is a block of code or collection of statements grouped together to perform a certain task or operation. The difference is that a method is associated with an object, while a function is not.

② In Java, methods are used to implement the behavior of the object.

③ A method must always be declared within a class.

④ A method can directly access all the instance variables of the enclosing class.

Method Definition in Java.

```
modifiers return_type method_name (parameters) exception_list
{
    // body of the method.
}
```

⇒ Method declarations have six components, in order:-

① Modifiers :- decides the visibility & accessibility of the method.

② The return type :- The data type of the value returned by the method, or void if the method does not return a value.

Date:

③ The method name :- identifier, name of the behavior

④ The parameter list in parenthesis :- a comma-separated list of input parameters, preceded by their data types. If there are no parameters, we must use empty parentheses.

⑤ An exception list :- list of exceptions ducked

⑥ The method body, enclosed between braces :- the method's code, activity / behavior exhibited by the object.

#Classification of Methods

Predefined Methods / Inbuilt Methods / Library Methods.

Eg :- println();
nextInt();
indexOf();
equals();

User - Defined Methods / Custom Methods

Eg :- eating();
writing();
adding();
guessNum();

[4 Types of Methods]

Type-1:

No Input

```
Void add()
```

No Output

Type-2:

No Input

```
int add()
```

Output

Type-3:

Input

```
void add(int a, int b)
```

No Output

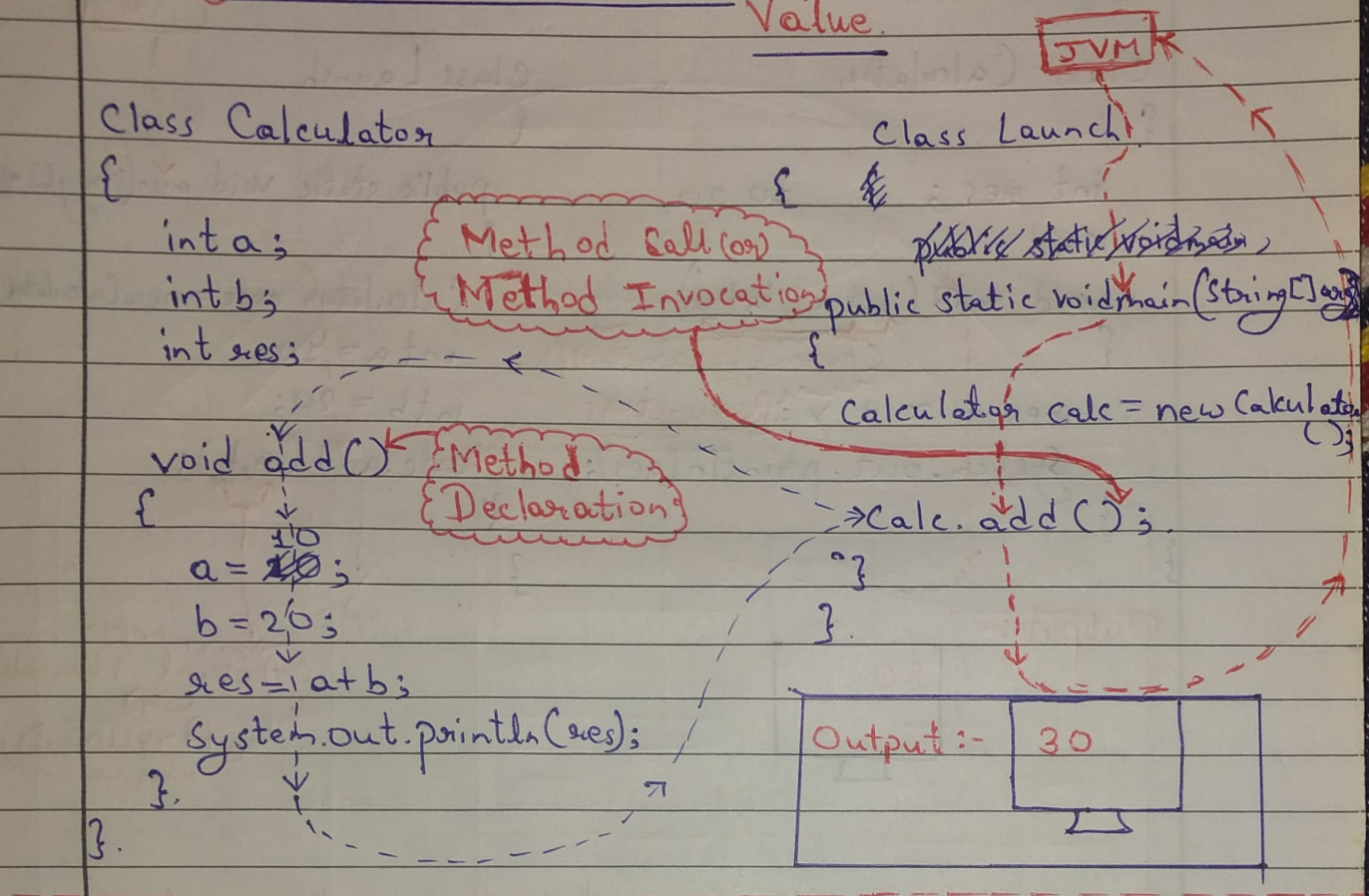
Type-4:

Input

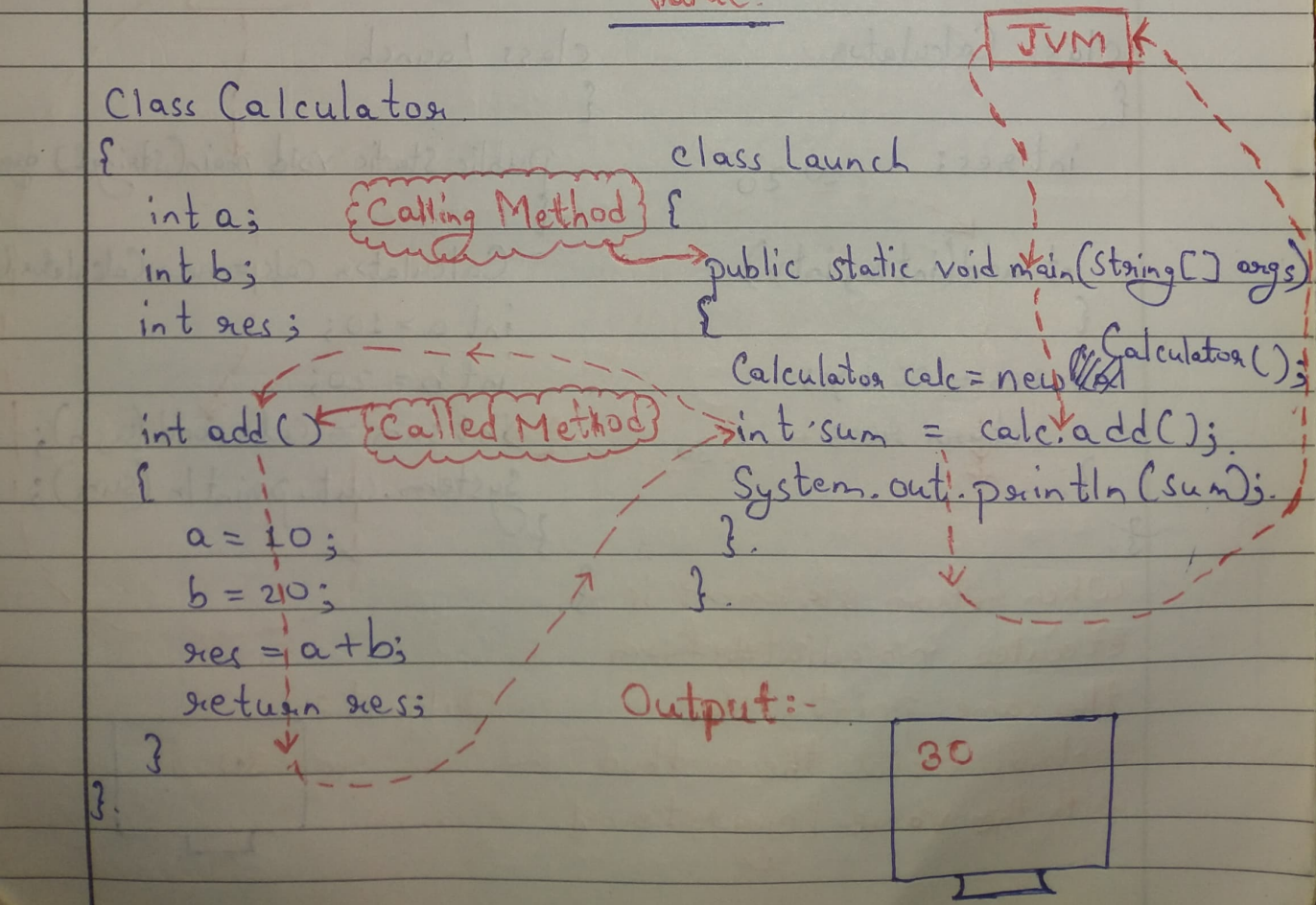
```
int add(int a, int b)
```

Output

Type-1: Method with no parameters & no return Value.



Type-2: Method with no parameters but with a return Value.



Type-3 : Method with parameters and no return value

```
Class Calculator
{
   int res ;
   void add (int x , int y)
   {
      res = x + y ;
      System.out.println(res);
   }
}
```

```
Class Launch
{
   public static void main(String[] args)
   {
      Calculator calc = new Calculator();
      int a = 10;
      int b = 20;
      calc.add (a,b);
   }
}
```

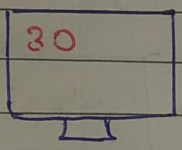
10 20.

Formal Parameters or Parameters

Actual Parameters or Arguments

JVM

Output :-



Type-4 : Method with parameters & a return value.

```
class Calculator
{
   int res;
   int add(int x, int y)
   {
      res = x + y;
      return res;
   }
}
```

```
class Launch
{
   public static void main(String[] args)
   {
      Calculator calc = new Calculator();
      int a = 10;
      int b = 20;
      int sum = calc.add (a, b);
      System.out.println (sum);
   }
}
```

10 20.

30.

JVM

When return statement is executed, immediately from the same point, the control exits the method with the value being returned.

Output :-