

METHODS IN JAVA

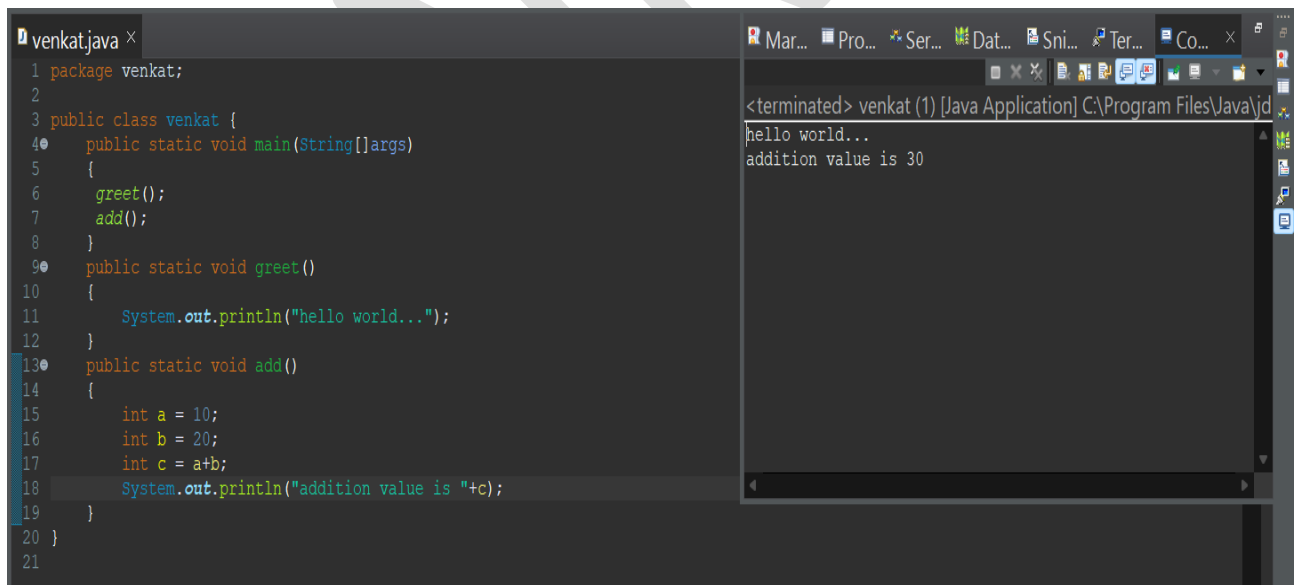
- ❖ Methods are the set of statements used to perform a parameter task
- ❖ If a method has to be executed , it has to be called by using its name

SYNTAX:

return type method name(parameter)

```
{  
    // definition i.e statements or body  
}
```

EX:



```
venkat.java x  
1 package venkat;  
2  
3 public class venkat {  
4     public static void main(String[] args)  
5     {  
6         greet();  
7         add();  
8     }  
9     public static void greet()  
10    {  
11        System.out.println("hello world...");  
12    }  
13    public static void add()  
14    {  
15        int a = 10;  
16        int b = 20;  
17        int c = a+b;  
18        System.out.println("addition value is "+c);  
19    }  
20 }  
21
```

The screenshot shows a Java IDE with a file named 'venkat.java'. The code defines a class 'venkat' with two methods: 'greet()' which prints 'hello world...' and 'add()' which calculates the sum of two integers 'a' (10) and 'b' (20), resulting in 'c' (30), and prints 'addition value is 30'. The output window on the right shows the execution results: '<terminated> venkat (1) [Java Application] C:\Program Files\Java\jd' followed by 'hello world...' and 'addition value is 30'.

TYPE ONE METHOD

Methods which would not accept any parameters and would not return any value.

EX:

The screenshot shows an IDE with two windows. The left window, titled 'venkat.java', contains the following Java code:

```
1 package venkat;  
2  
3 public class venkat {  
4     public static void main(String[] args)  
5     {  
6         add();  
7     }  
8     public static void add()  
9     {  
10        int a=10;  
11        int b=20;  
12        int c =a+b;  
13        System.out.println("addition is "+c);  
14    }  
15 }  
16
```

The right window, titled '<terminated> venkat (1) [Java Application] C:\Program Files\Java\jdk...', displays the output of the program:

```
addition is 30
```

TYPE TWO METHOD

Methods which would not accept any parameters and would return a value.

EX:

The screenshot shows an IDE with two windows. The left window, titled 'venkat.java', contains the following Java code:

```
1 package venkat;
2
3 public class venkat {
4     public static void main(String[] args)
5     {
6         int res = sub();
7     }
8     public static int sub()
9     {
10        int a=20;
11        int b=10;
12        int c =a-b;
13        System.out.println("subtraction value is "+c);
14        return c;
15    }
16 }
17
```

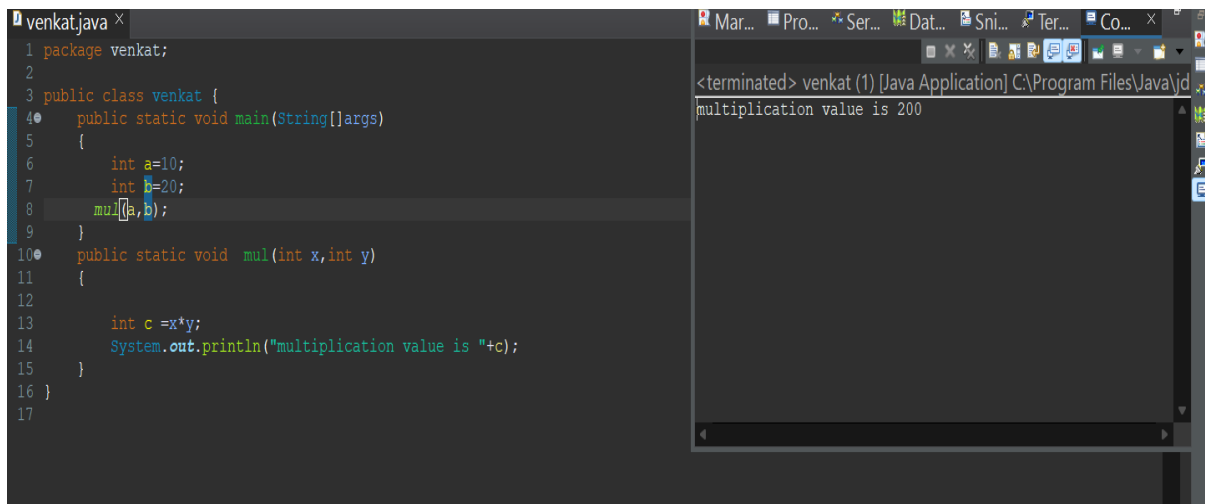
The right window, titled 'Mar... Pro... Ser... Dat... Sni... Ter... Co...', shows the output of the program:

```
<terminated> venkat (1) [Java Application] C:\Program Files\Java\
subtraction value is 10
```

TYPE THREE METHOD

Methods which would accept parameters and would not return any value.

EX:



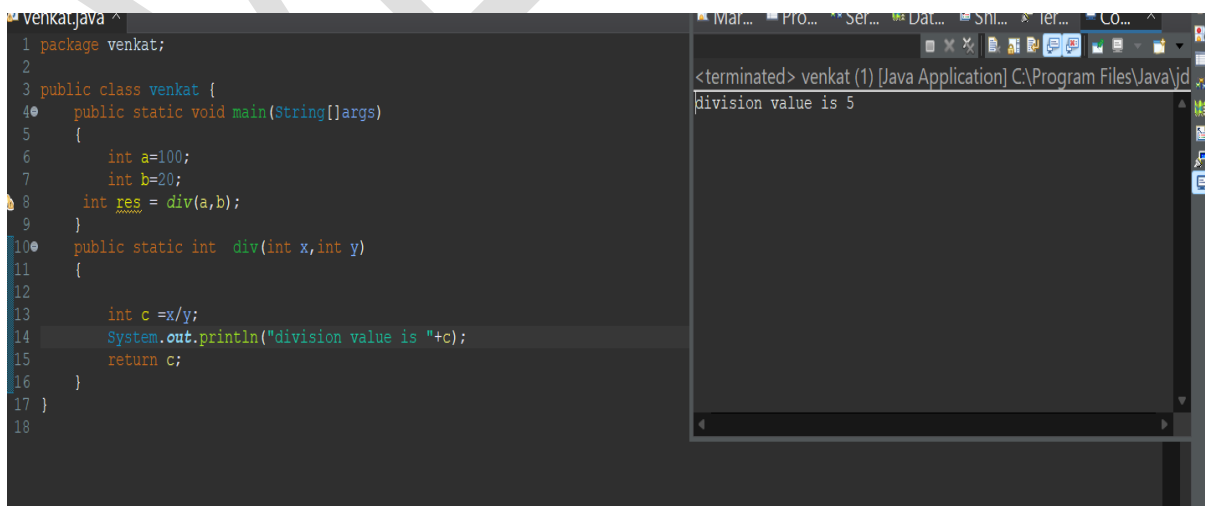
The screenshot shows a Java IDE with a file named 'venkat.java'. The code defines a package 'venkat' and a class 'venkat'. Inside the class, there is a 'main' method that takes a String array 'args' as a parameter. It declares two integer variables 'a' and 'b' with values 10 and 20 respectively, and then calls a method 'mul(a,b)'. The 'mul' method is a static void method that takes two integer parameters 'x' and 'y', calculates their product 'c = x*y', and prints the result using 'System.out.println("multiplication value is "+c);'. The output window on the right shows the message 'multiplication value is 200'.

```
1 package venkat;
2
3 public class venkat {
4     public static void main(String[] args)
5     {
6         int a=10;
7         int b=20;
8         mul(a,b);
9     }
10    public static void mul(int x,int y)
11    {
12
13        int c =x*y;
14        System.out.println("multiplication value is "+c);
15    }
16 }
17
```

TYPE FOUR METHOD

Methods which would accept parameters and would return any value.

EX:



The screenshot shows a Java IDE with a file named 'venkat.java'. The code defines a package 'venkat' and a class 'venkat'. Inside the class, there is a 'main' method that takes a String array 'args' as a parameter. It declares two integer variables 'a' and 'b' with values 100 and 20 respectively, and then calls a method 'div(a,b)' which returns an integer value. The 'div' method is a static int method that takes two integer parameters 'x' and 'y', calculates their quotient 'c = x/y', and returns the result. The output window on the right shows the message 'division value is 5'.

```
1 package venkat;
2
3 public class venkat {
4     public static void main(String[] args)
5     {
6         int a=100;
7         int b=20;
8         int res = div(a,b);
9     }
10    public static int div(int x,int y)
11    {
12
13        int c =x/y;
14        System.out.println("division value is "+c);
15        return c;
16    }
17 }
18
```

BY VENKATADRI V

VENKAT