

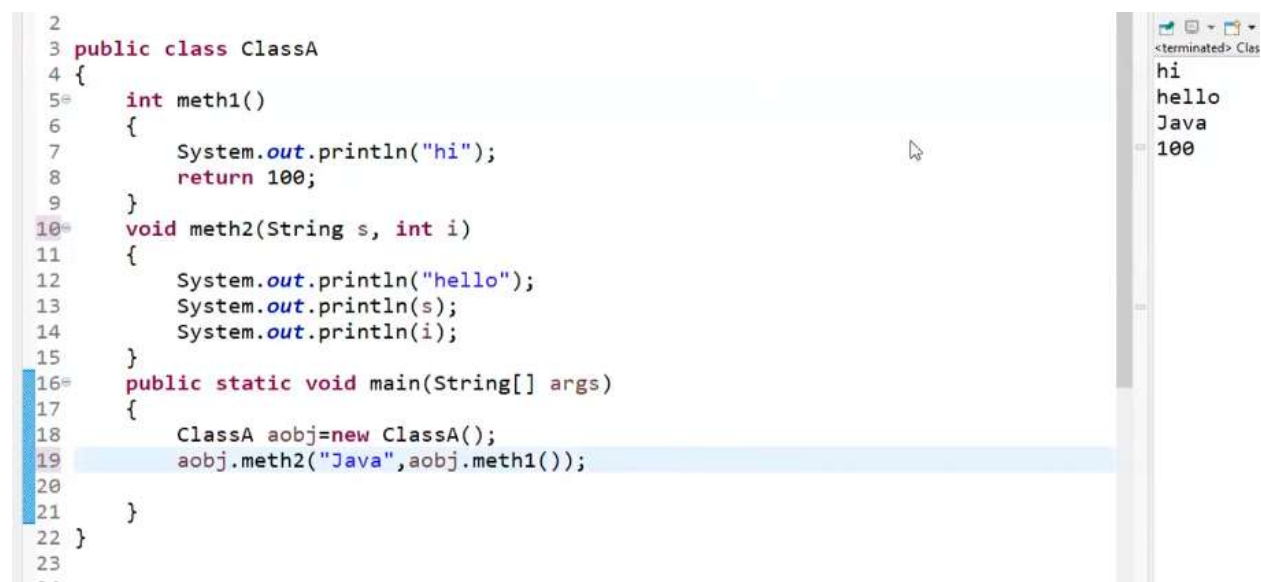
Q) How to pass values for the method parameters?

A) We need to initialize a method parameter whenever we are calling a method.

Note:

The value which is returned by a method we can use that value in below scenarios.

1. We can print that value.
2. We can store that value in its respective data types.
3. We can use that value in some arithmetical operations.
4. We can return that value from another method.
5. We can use that value as a parameter for another method.



```
2
3 public class ClassA
4 {
5     int meth1()
6     {
7         System.out.println("hi");
8         return 100;
9     }
10    void meth2(String s, int i)
11    {
12        System.out.println("hello");
13        System.out.println(s);
14        System.out.println(i);
15    }
16    public static void main(String[] args)
17    {
18        ClassA aobj=new ClassA();
19        aobj.meth2("Java",aobj.meth1());
20    }
21 }
22
23
```

The screenshot shows a Java IDE with a class named ClassA. The class has two methods: meth1() which prints "hi" and returns 100, and meth2(String s, int i) which prints "hello", the string s, and the integer i. The main method creates an instance of ClassA and calls meth2("Java", aobj.meth1()), demonstrating how the return value of one method is passed as an argument to another.

```

ClassA.java x ClassB.java
5= int meth1()
6 {
7     System.out.println("meth1() called");
8     return 100;
9 }
10= int meth2(int x)
11 {
12     System.out.println("meth2() called");
13     return x+10; // 100+10 ==> 110
14 }
15= int meth3(int a, int b) // a=50 b=110
16 {
17     System.out.println("Java is awesome");
18     System.out.println("==>"+(a+b));
19     return 50;
20 }
21= public static void main(String[] args)
22 {
23     System.out.println(new ClassA().meth3(new ClassA().meth1()/2, new ClassA(
24 }
25 }
26
27

```

Console x

```

<terminated> ClassA [Java Application]
meth1() called
meth1() called
meth2() called
Java is awesome
==>160
50

```

```

3 public class ClassA
4 {
5= int meth1(int a, int b)
6 {
7     System.out.println(a);
8     System.out.println(a-b);
9     return new ClassA().meth4(a-1,"Hi");
10 }
11= String meth2(int x)
12 {
13     System.out.println(x-new ClassA().meth1(200, 199));
14     System.out.println(x);
15     return "Java ";
16 }
17= String meth3(int a, int b, String s)
18 {
19     System.out.println(a);
20     System.out.println(a+a);
21     System.out.println(a+b);
22     return s;
23 }

```

1 changed line

```

16     }
17     String meth3(int a, int b, String s)
18     {
19         System.out.println(a);
20         System.out.println(a+a);
21         System.out.println(a+b);
22         return s;
23     }
24     int meth4(int a, String s)
25     {
26         System.out.println(a);
27         System.out.println(s);
28         return a+a;
29     }
30     String meth5(String s)
31     {
32         return s;
33     }
34     public static void main(String[] args)
35     {
36         System.out.println(new ClassA().meth5(new ClassA().meth3(15, 25, new ClassA().meth2(500))));
37     }
38 }

```

The screenshot shows an IDE with two tabs: ClassA.java and ClassB.java. The code in ClassA.java is the same as shown in the first block. The console window on the right shows the output of the program, which is the result of the nested method calls in the main method. The output is as follows:

```

<terminated> ClassA [Java Application] C:\Pro
200
1
199
Hi
102
500
15
30
40
Java

```

```
2
3 public class ClassA
4 {
5     public String testMethod()
6     {
7         System.out.println(35);
8         return new ClassA().testMethod3(new ClassA().testMethod2()) + new ClassA().testMethod4(" here");
9     }
10    public int testMethod2()
11    {
12        System.out.println(25);
13        return 25+5;
14    }
15    public String testMethod3(int a)
16    {
17        System.out.println(15);
18        return "is";
19    }
20    public String testMethod4(String s)
21    {
22        System.out.println(45);
23        return " awesome";
24    }
25    public static void main(String[] args)
26    {
```

```
6
7        System.out.println(35);
8        return new ClassA().testMethod3(new ClassA().testMethod2()) + new ClassA().testMethod4(" here");
9    }
10    public int testMethod2()
11    {
12        System.out.println(25);
13        return 25+5;
14    }
15    public String testMethod3(int a)
16    {
17        System.out.println(15);
18        return "is";
19    }
20    public String testMethod4(String s)
21    {
22        System.out.println(45);
23        return " awesome";
24    }
25    public static void main(String[] args)
26    {
27        ClassA t = new ClassA();
28        System.out.println("Java " + t.testMethod());
29    }
30 }
```