INDIAN INSTITUTE OF TECHNOLOGY ROORKEE

Fundamentals of Object Oriented Programming

CSN- 103

Dr. R. Balasubramanian

Associate Professor

Department of Computer Science and Engineering

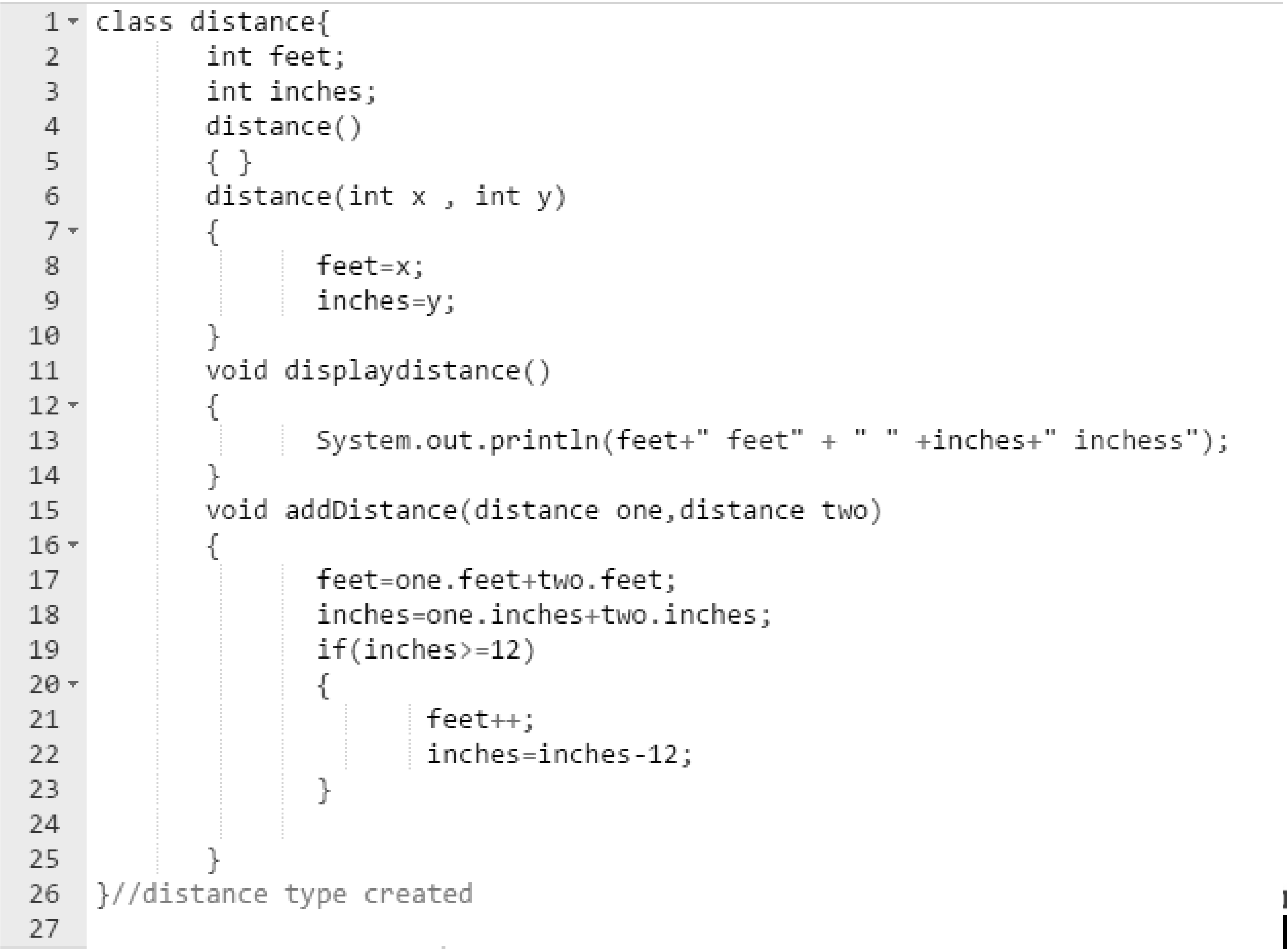
Indian Institute of Technology Roorkee

Roorkee 247 667

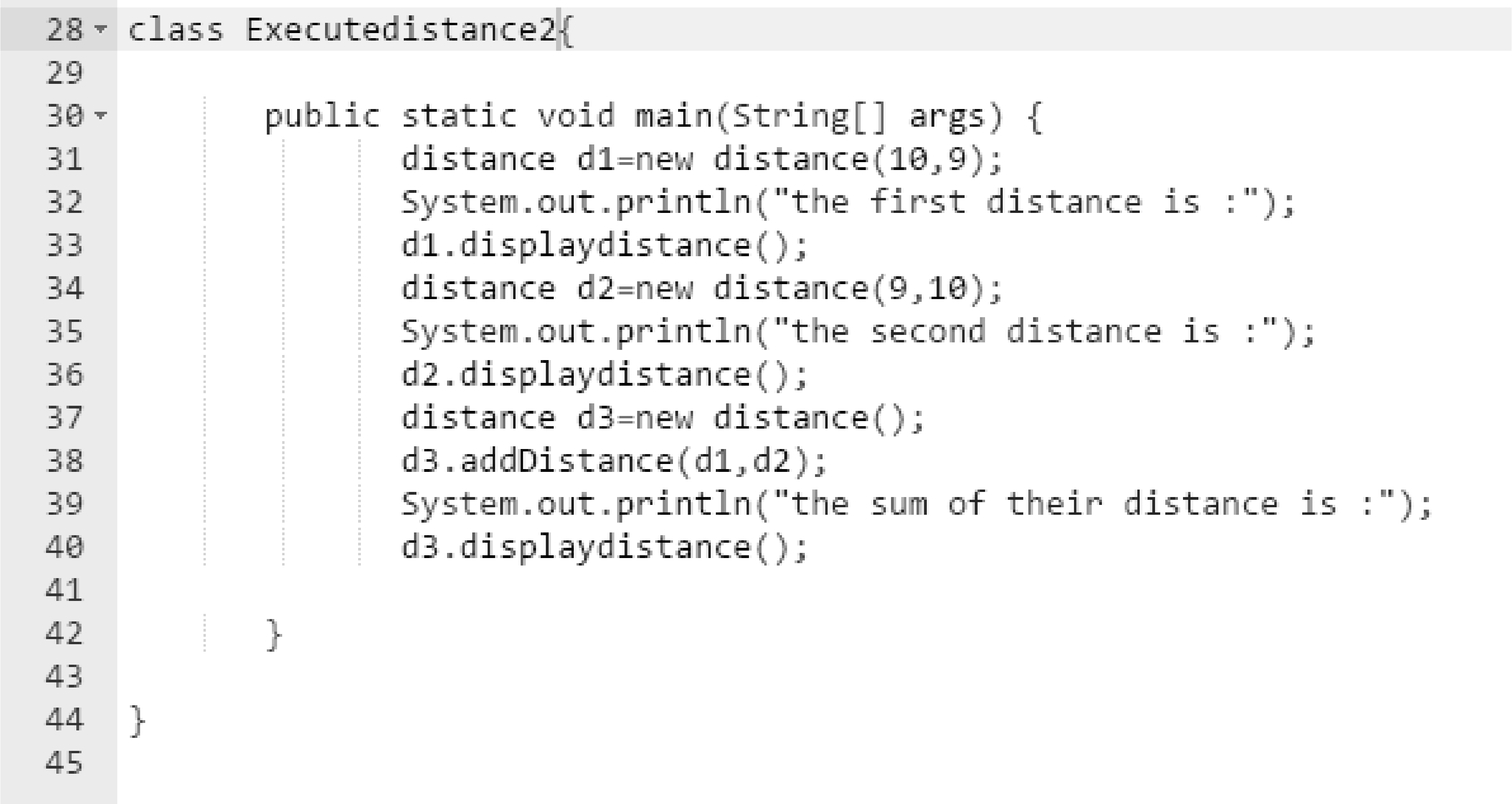
balarfcs@iitr.ac.in

https://sites.google.com/site/balaiitr/

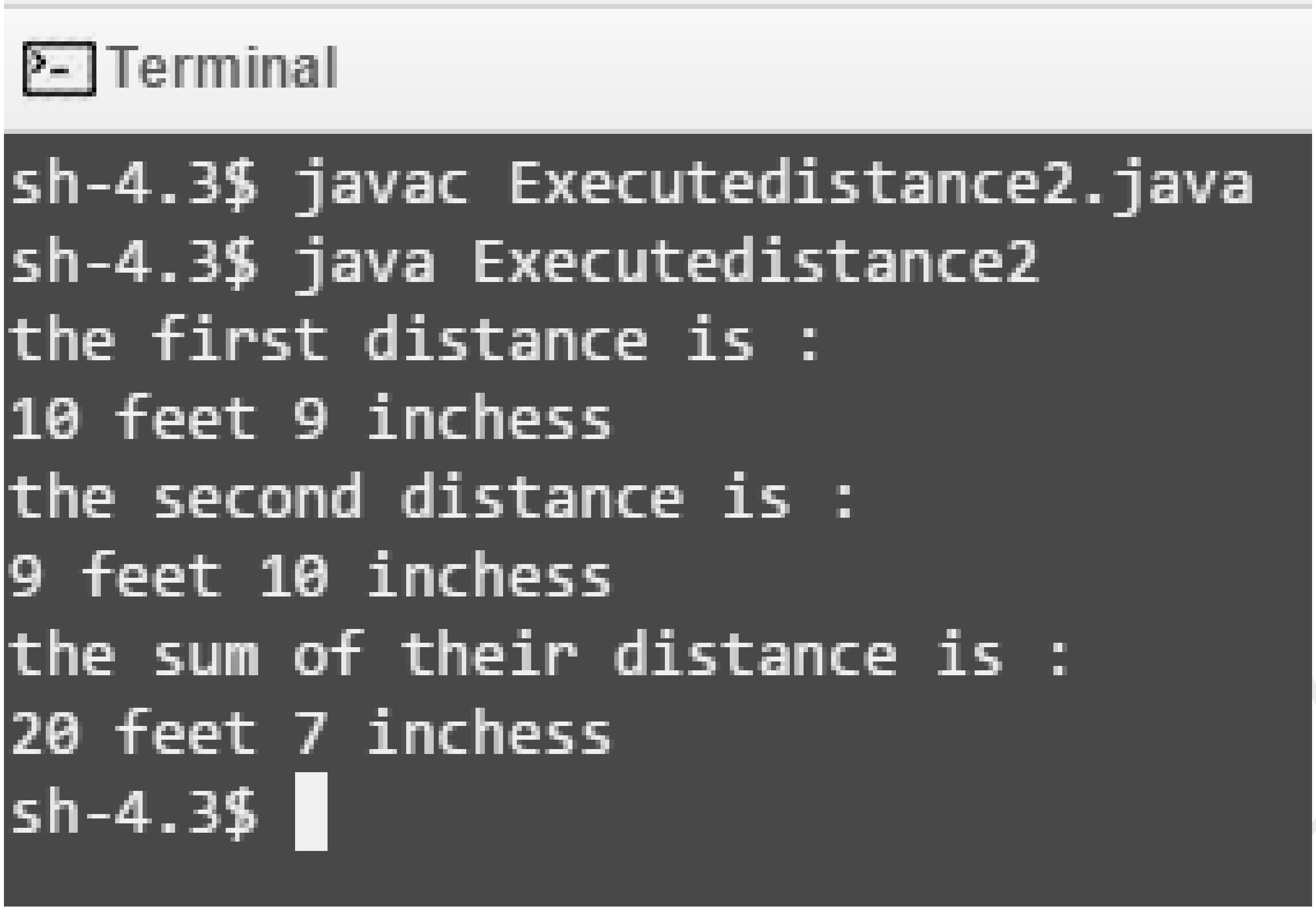
Objects as Arguments



```
class distance{
        int feet;
        int inches;
        distance()
        { }
        distance(int x , int y)
        {
                feet=x;
                inches=y;
        }
        void displaydistance()
        {
                System.out.println(feet+" feet" + " " +inches+" inchess");
        }
        void addDistance(distance one,distance two)
        {
                feet=one.feet+two.feet;
                inches=one.inches+two.inches;
                if(inches>=12)
                {
                        feet++;
                        inches=inches-12;
                }

        }
}//distance type created

```

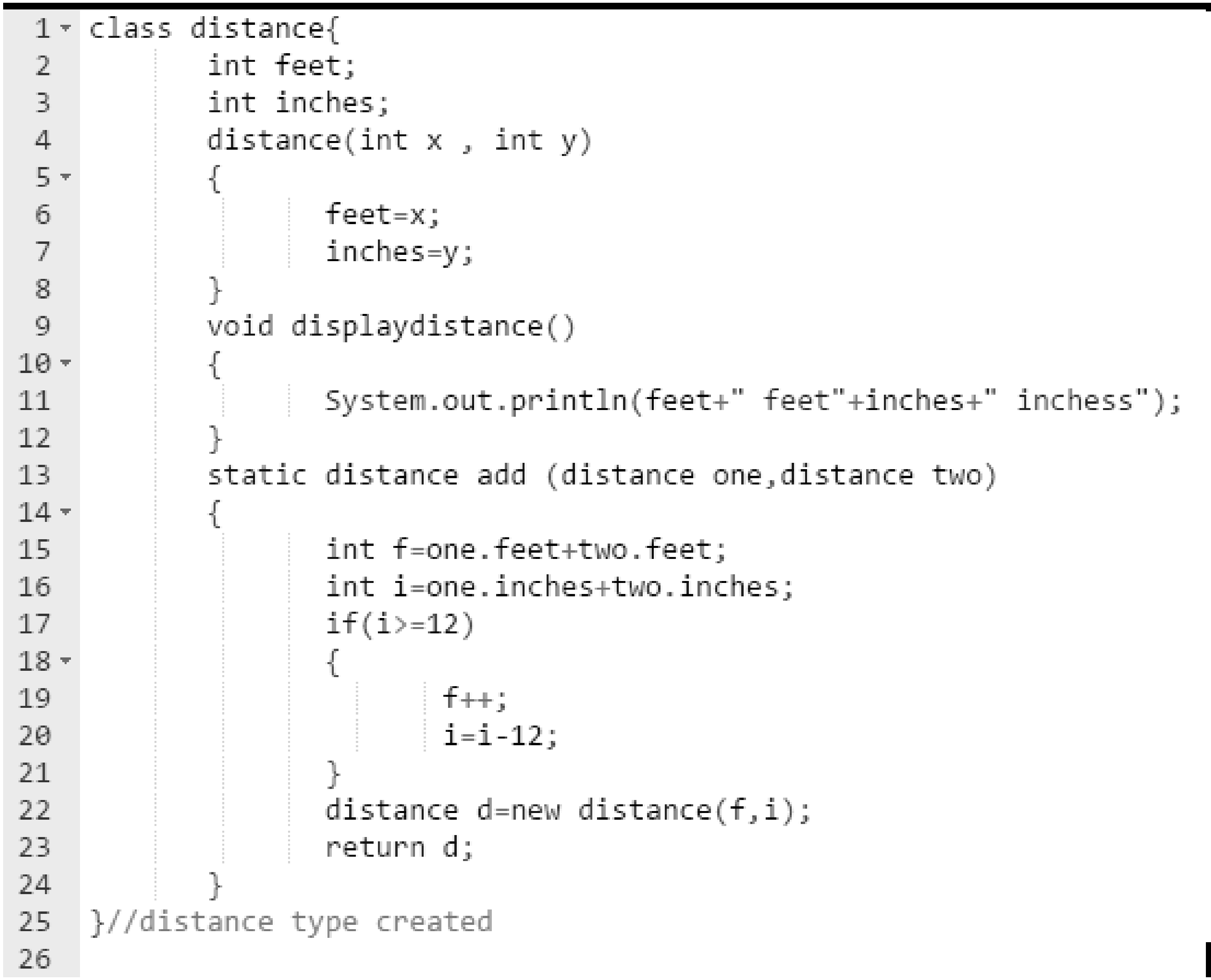


```java
class Executedistance2{

	public static void main(String[] args) {
			distance d1=new distance(10,9);
			System.out.println("the first distance is :");
			d1.displaydistance();
			distance d2=new distance(9,10);
			System.out.println("the second distance is :");
			d2.displaydistance();
			distance d3=new distance();
			d3.addDistance(d1,d2);
			System.out.println("the sum of their distance is :");
			d3.displaydistance();

	}

}
```

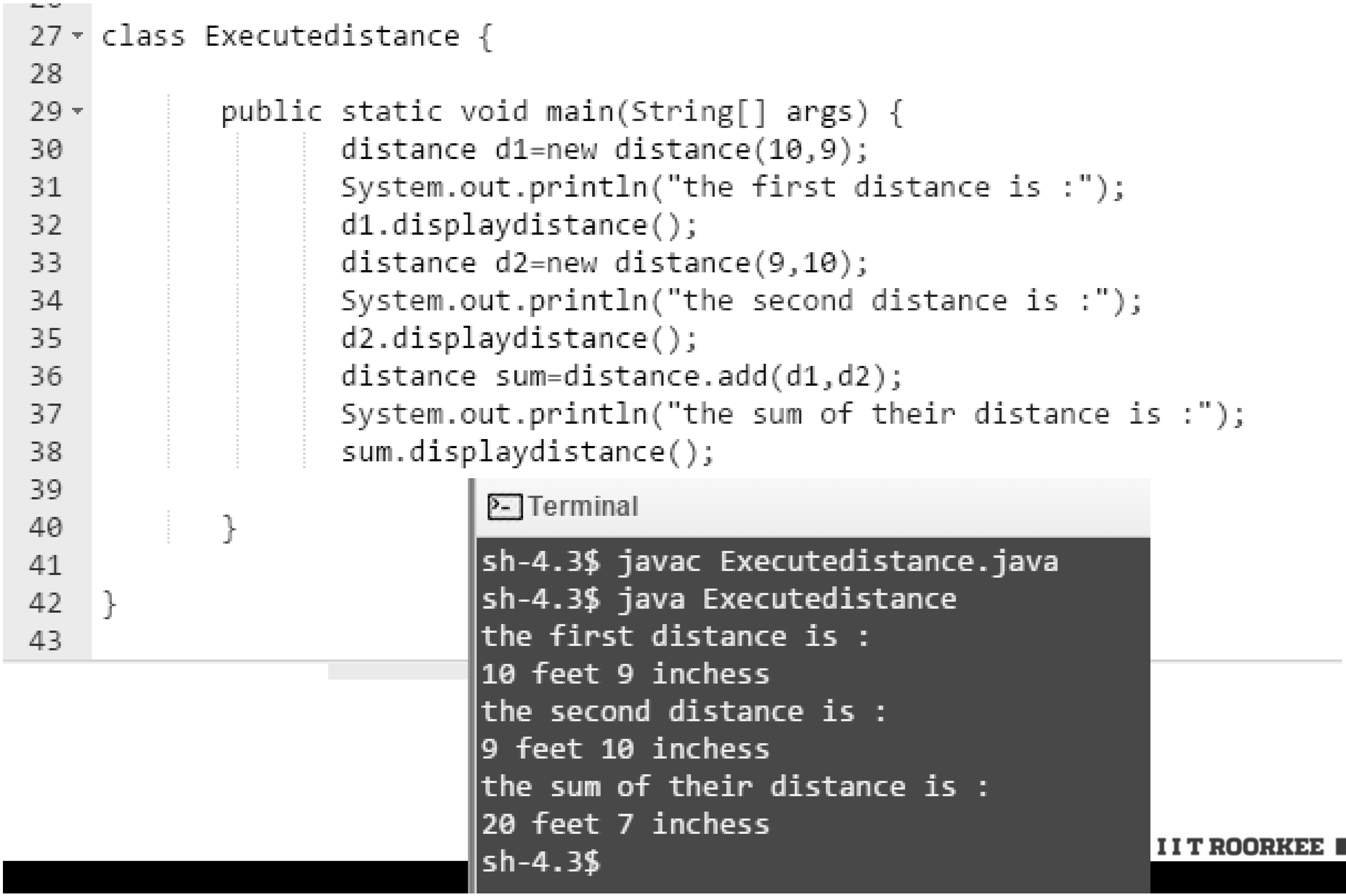


```
sh-4.3$ javac Executedistance2.java
sh-4.3$ java Executedistance2
the first distance is :
10 feet 9 inchess
the second distance is :
9 feet 10 inchess
the sum of their distance is :
20 feet 7 inchess
sh-4.3$
```

Method returning Objects



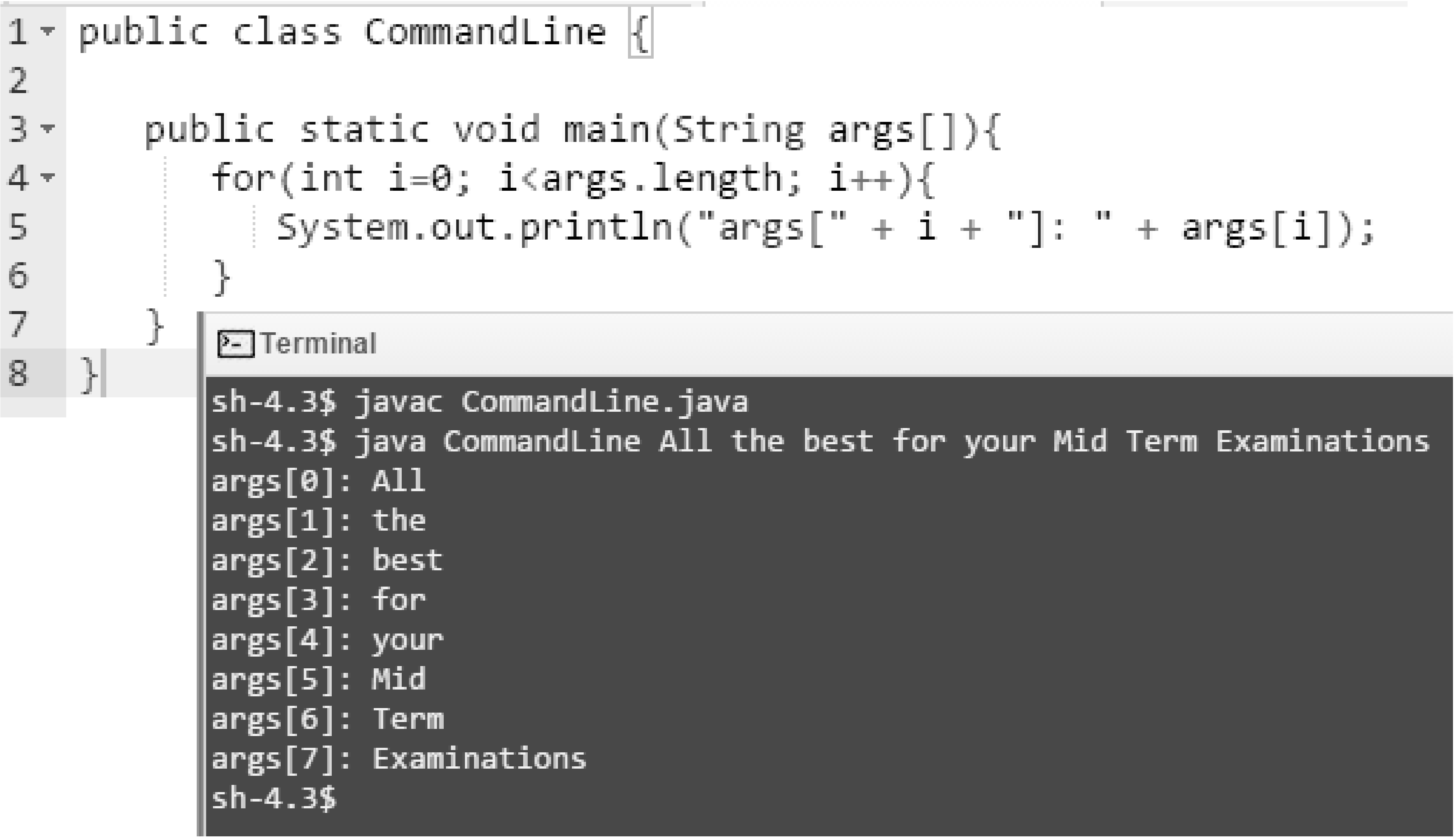
```
class distance{
        int feet;
        int inches;
        distance(int x , int y)
        {
                feet=x;
                inches=y;
        }
        void displaydistance()
        {
                System.out.println(feet+" feet"+inches+" inchess");
        }
        static distance add (distance one,distance two)
        {
                int f=one.feet+two.feet;
                int i=one.inches+two.inches;
                if(i>=12)
                {
                        f++;
                        i=i-12;
                }
                distance d=new distance(f,i);
                return d;
        }
}//distance type created
```



```java
class Executedistance {

        public static void main(String[] args) {
                distance d1=new distance(10,9);
                System.out.println("the first distance is :");
                d1.displaydistance();
                distance d2=new distance(9,10);
                System.out.println("the second distance is :");
                d2.displaydistance();
                distance sum=distance.add(d1,d2);
                System.out.println("the sum of their distance is :");
                sum.displaydistance();

        }

}
```

```
sh-4.3$ javac Executedistance.java
sh-4.3$ java Executedistance
the first distance is :
10 feet 9 inchess
the second distance is :
9 feet 10 inchess
the sum of their distance is :
20 feet 7 inchess
sh-4.3$
```

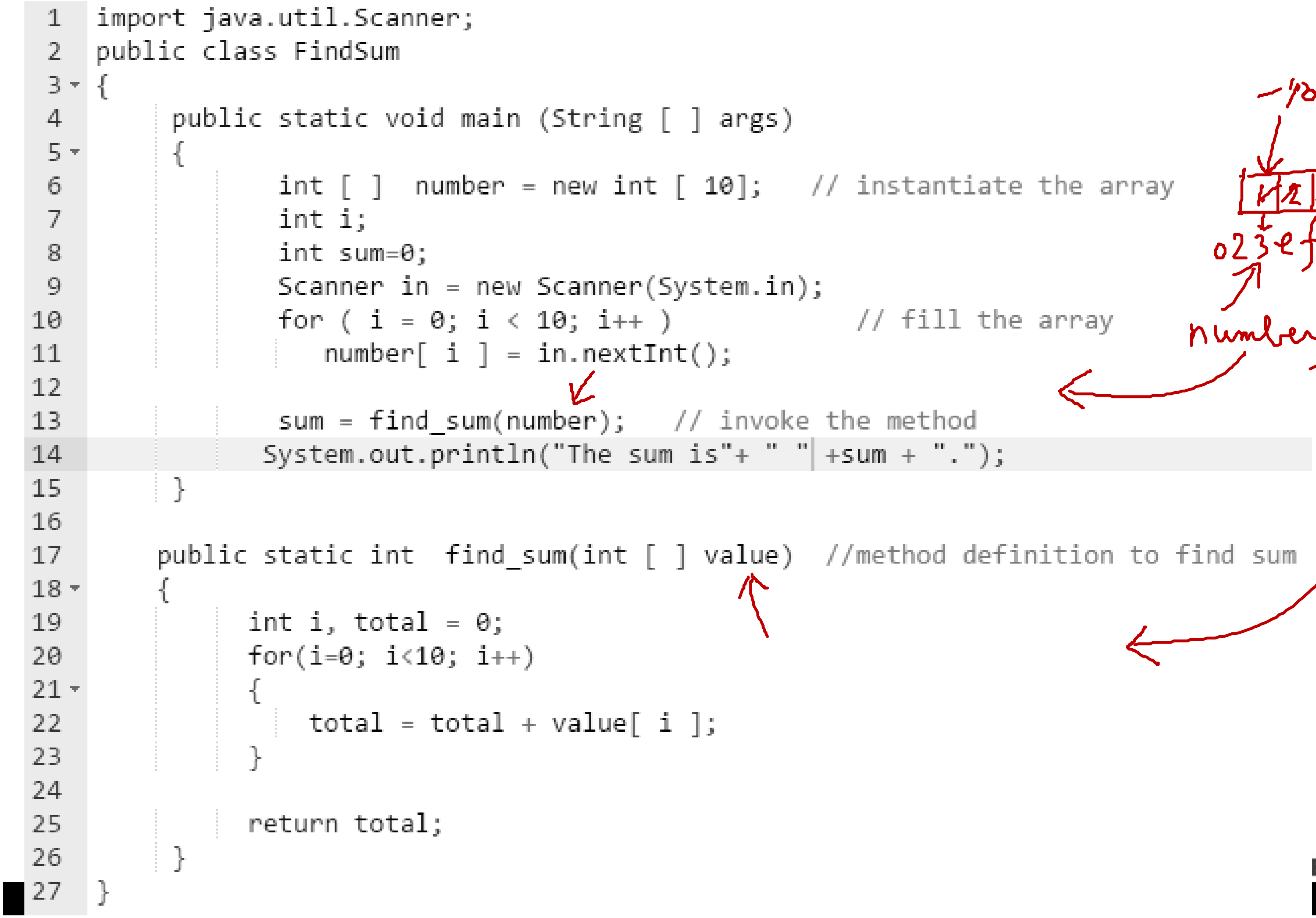
Using Command-Line Arguments



```java
public class CommandLine {

    public static void main(String args[]){
        for(int i=0; i<args.length; i++){
            System.out.println("args[" + i + "]: " + args[i]);
        }
    }
}
```

```
sh-4.3$ javac CommandLine.java
sh-4.3$ java CommandLine All the best for your Mid Term Examinations
args[0]: All
args[1]: the
args[2]: best
args[3]: for
args[4]: your
args[5]: Mid
args[6]: Term
args[7]: Examinations
sh-4.3$
```

Passing Arrays to Methods



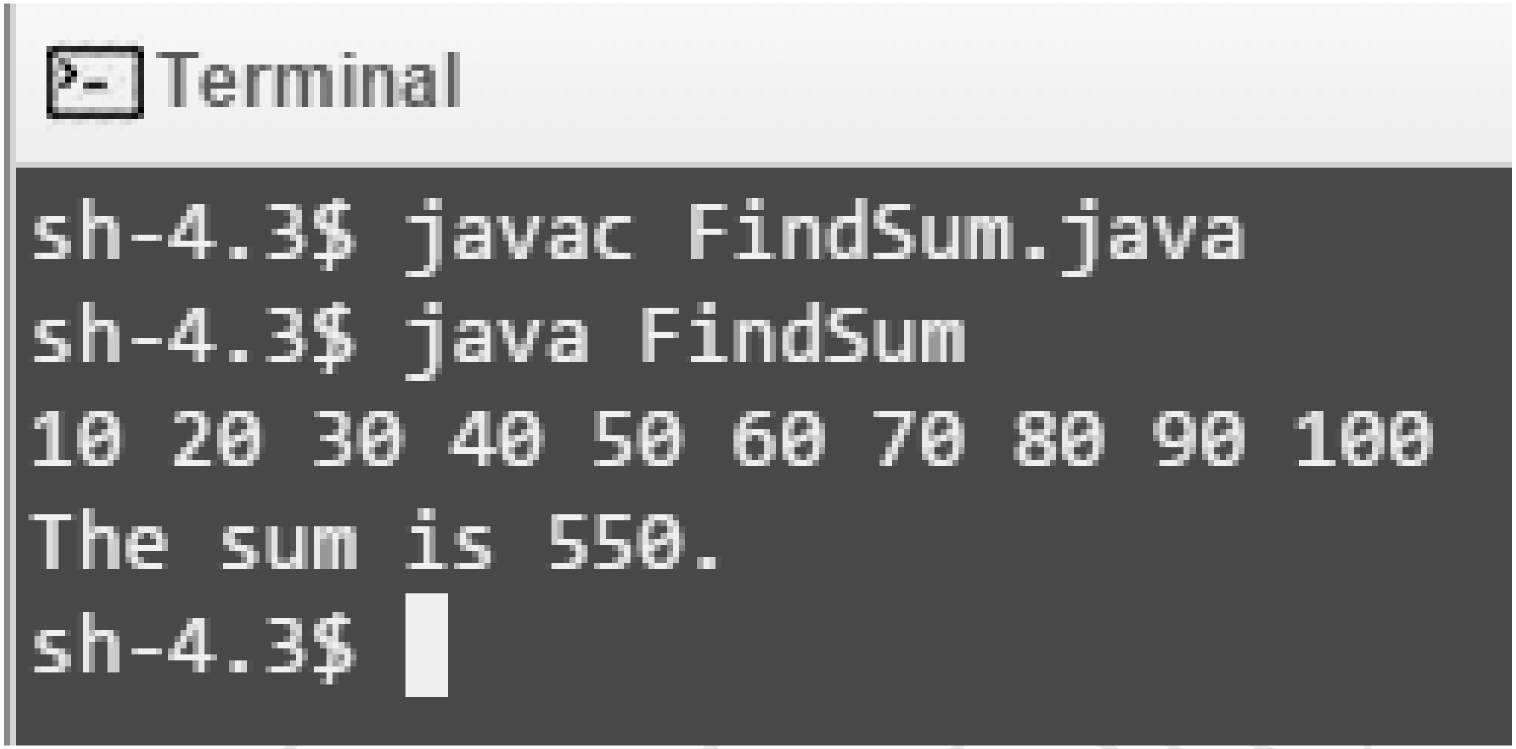
```
import java.util.Scanner;
public class FindSum
{
     public static void main (String [ ] args)
     {
          int [ ]  number = new int [ 10];   // instantiate the array
          int i;
          int sum=0;
          Scanner in = new Scanner(System.in);
          for ( i = 0; i < 10; i++ )             // fill the array
             number[ i ] = in.nextInt();

          sum = find_sum(number);   // invoke the method
         System.out.println("The sum is"+ " " +sum + ".");
     }

    public static int  find_sum(int [ ] value)  //method definition to find sum
    {
          int i, total = 0;
          for(i=0; i<10; i++)
          {
              total = total + value[ i ];
          }

          return total;
     }
}
```

-10 -10 -10

1 2 3 . . .

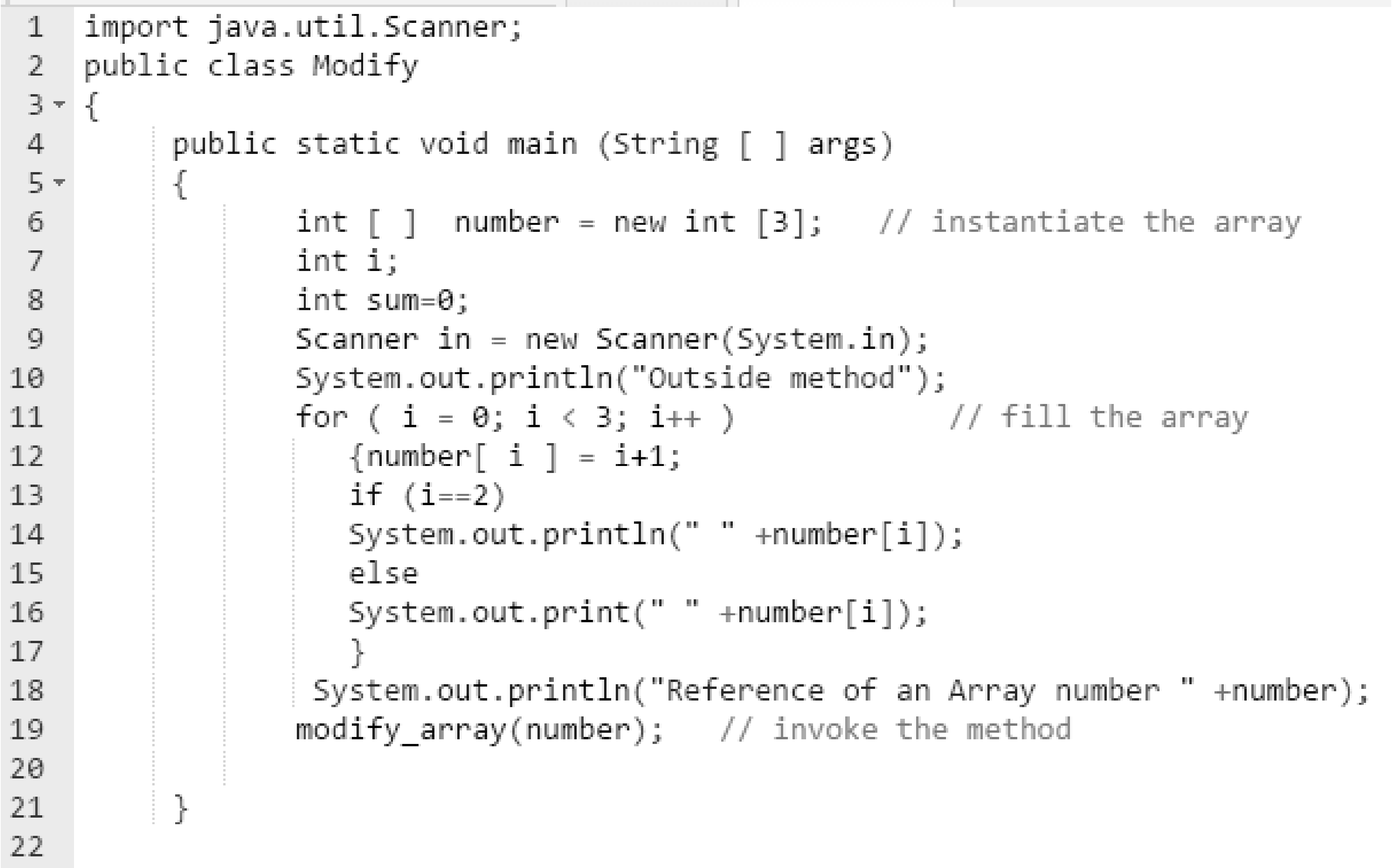
023ef

number, value

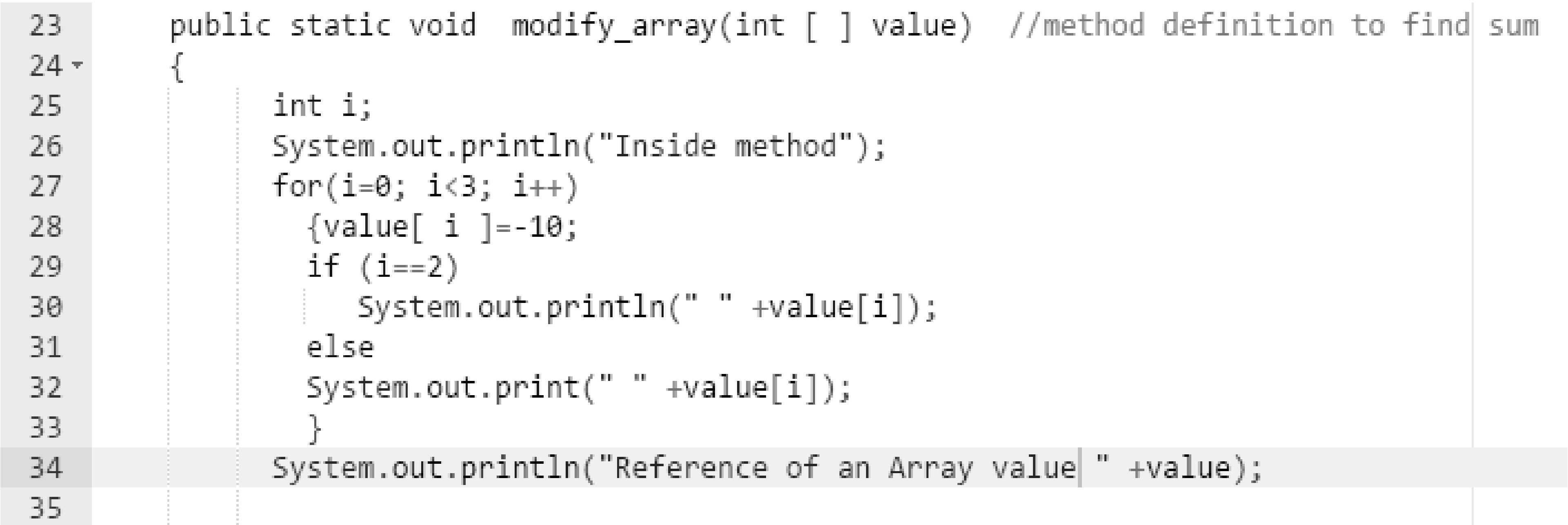


```
sh-4.3$ javac FindSum.java
sh-4.3$ java FindSum
10 20 30 40 50 60 70 80 90 100
The sum is 550.
sh-4.3$ 
```

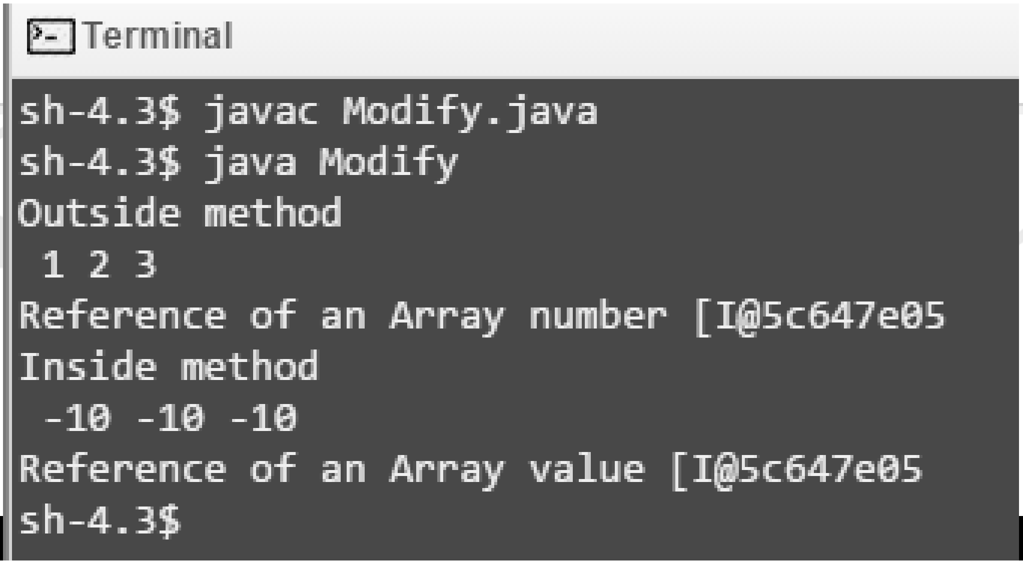
Java passes reference by value



```
import java.util.Scanner;
public class Modify
{
     public static void main (String [ ] args)
     {
            int [ ]  number = new int [3];   // instantiate the array
            int i;
            int sum=0;
            Scanner in = new Scanner(System.in);
            System.out.println("Outside method");
            for ( i = 0; i < 3; i++ )               // fill the array
               {number[ i ] = i+1;
               if (i==2)
               System.out.println(" " +number[i]);
               else
               System.out.print(" " +number[i]);
               }
             System.out.println("Reference of an Array number " +number);
            modify_array(number);   // invoke the method

     }

```

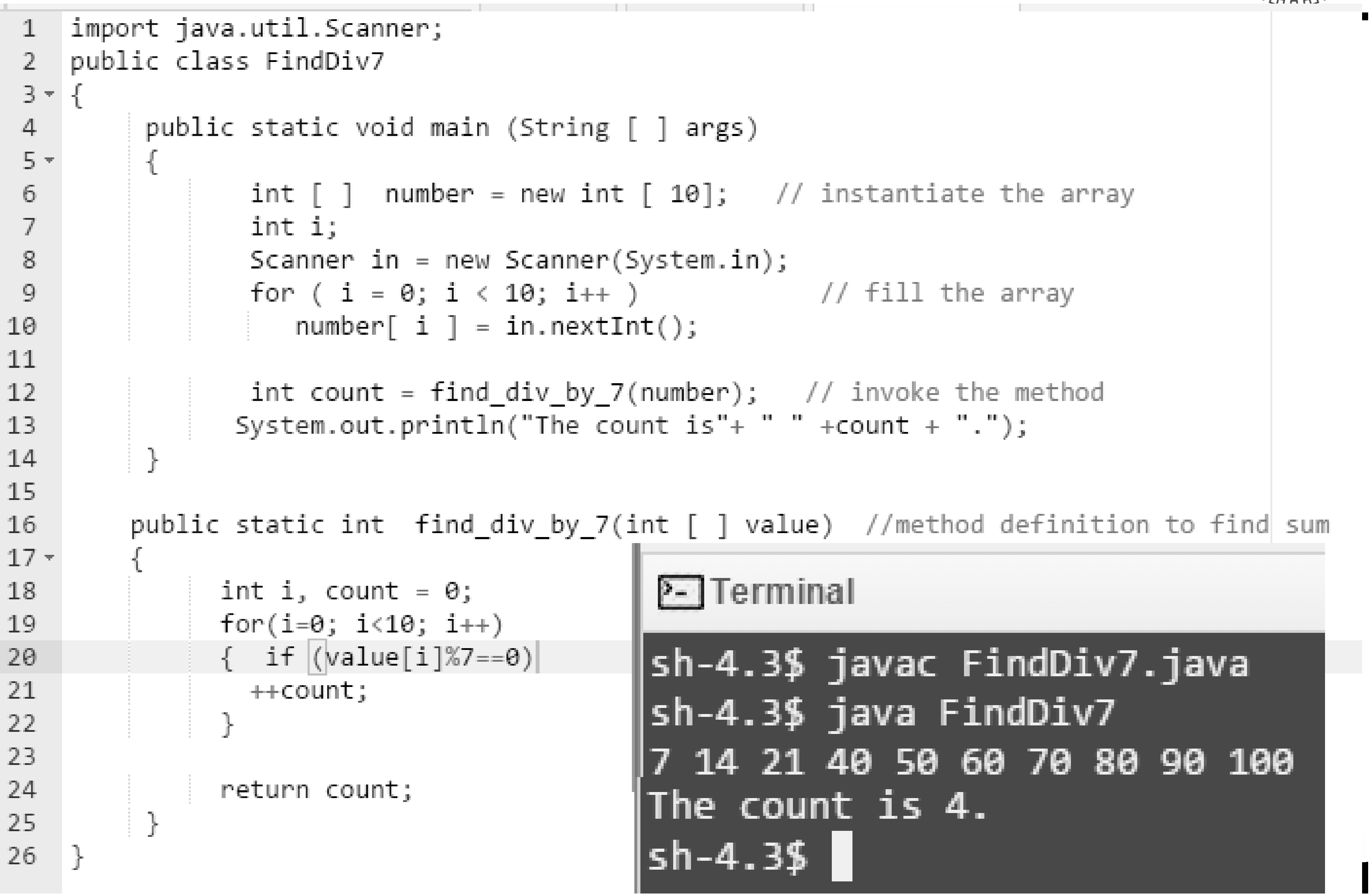


```
    public static void  modify_array(int [ ] value)  //method definition to find sum
    {
          int i;
          System.out.println("Inside method");
          for(i=0; i<3; i++)
            {value[ i ]=-10;
            if (i==2)
               System.out.println(" " +value[i]);
            else
            System.out.print(" " +value[i]);
            }
          System.out.println("Reference of an Array value " +value);

     }
}
```



```
sh-4.3$ javac Modify.java
sh-4.3$ java Modify
Outside method
 1 2 3
Reference of an Array number [I@5c647e05
Inside method
 -10 -10 -10
Reference of an Array value [I@5c647e05
sh-4.3$
```

WAP using a method to find the number of elements in a given integer array which are divisible by 7. Pass this array in the method.



```
import java.util.Scanner;
public class FindDiv7
{
     public static void main (String [ ] args)
     {
            int [ ]  number = new int [ 10];   // instantiate the array
            int i;
            Scanner in = new Scanner(System.in);
            for ( i = 0; i < 10; i++ )             // fill the array
               number[ i ] = in.nextInt();

            int count = find_div_by_7(number);   // invoke the method
           System.out.println("The count is"+ " " +count + ".");
     }

    public static int  find_div_by_7(int [ ] value)  //method definition to find sum
    {
          int i, count = 0;
          for(i=0; i<10; i++)
          {  if (value[i]%7==0)
            ++count;
          }

          return count;
     }
}
```

Terminal

```
sh-4.3$ javac FindDiv7.java
sh-4.3$ java FindDiv7
7 14 21 40 50 60 70 80 90 100
The count is 4.
sh-4.3$
```

JAVA

- **Java is pass by value**. Well, **pass by reference value**.

- Oh well, even better is ***pass-by-copy-of-the-variable-value! ;)***