

## EXAMPLE PROGRAM • ARRAYS AS PARAMETERS

```
public class ArrayOps
{
    public ArrayOps() { }

    // Copy source to target
    public void copy (int[] source, int[] target)
    {
        for (int count=0; count < source.length; count++)
            target[ count ] = source[ count ];
    }

    public void printList (int[] list)
    /* list is a reference parameter. list is the same array as
       array data in the main method */
    {
        for (int index = 0; index < list.length; index++)
            System.out.print( list[index] + "  ");

        System.out.println();
        System.out.println();
    }

    public void squareList (int[] list)
    /* Array list is a local alias for array data in the main method. Any
       reference to local list is a reference to array data in function main. */
    {
        for (int index = 0; index < list.length; index++)
            list[index] = list[index] * list[index];
    }

    public void rotateList (int[] list)
    /* This method is working with a local copy of the array passed
       as an argument. Changes to local array list will have no effect
       on the array data in the calling method. This method will shift each
       value one cell to the right. The value in list[list.length-1] will be
       moved to list[0]. Before the method is completed, printList will
       be called. The point of this method is to illustrate an array as
       a value parameter. */
    {
        int temp = list[list.length-1];

        int[] listCopy = new int[ list.length ];

        copy(list, listCopy);

        for (int loop = listCopy.length-1; loop > 0; loop--)
            listCopy[loop] = listCopy[loop-1];
        listCopy[0] = temp;

        System.out.println("Inside of rotateList: ");
        printList (listCopy);
    }
}
```

```

public static void main (String[] args)
{
    final int MAX = 6;

    ArrayOps arrayOps = new ArrayOps();

    int[] data = new int[MAX];

    for (int loop = 0; loop < MAX; loop++)
        data[loop] = loop;    // initialize array

    System.out.println("Array initialized:  ");

    arrayOps.printList(data);    // print array in ascending order
    arrayOps.squareList (data);
    System.out.println("Array after call of squareList:  ");
    arrayOps.printList (data);
    arrayOps.rotateList (data);
    System.out.println("Array after call of rotateList:  ");
    arrayOps.printList (data);    // print list again
}
}

```