

Use an array in a method

```
public int nthFib(int n) {
    if(n <= 1)
        return n;
    int[] fibs = new int[n];
    fibs[0] = 0;
    fibs[1] = 1;
    for(int i=2; i<n; i++)
        fibs[i] = fibs[i-1] + fibs[i-2];
    return fibs[n-1];
}
```

Pass an array into a method

```
public void foo() {
    int[] points = new int[4];
    points[0] = 2;
    points[1] = 3;
    points[2] = 5;
    points[3] = 7;
    double dist = distance(points);
}

public double distance(int[] coords) {
    double xDist = coords[2] - coords[0];
    double yDist = coords[3] - coords[1];
    return Math.sqrt(xDist*xDist + yDist*yDist);
}
```

Return an array from a method

```
public void foo() {
    double[] roots = quadFormula(1, 5, 6);
    for(int i = 0; i<roots.length; i++)
        System.out.println(roots[i]);
}

public double[] quadFormula(int a, int b, int c) {
    int discr = b*b - 4*a*c;
    if(discr < 0)
        return new double[0];
    else if(discr == 0)
        return new double[]{-b/(2.0*a)};
    else {
        double[] roots = new double[2];
        roots[0] = (-b + Math.sqrt(discr))/(2.0*a);
        roots[1] = (-b - Math.sqrt(discr))/(2.0*a);
        return roots;
    }
}
```

Pass an array to a method... and MODIFY IT!

```
public void foo() {
    int[] array = new int[5];
    fill(array);
    for(int i=0; i<array.length; i++)
        System.out.println(array[i]);
}

public void fill(int[] theArray) {
    for(int i=0; i<theArray.length; i++)
        theArray[i] = i;
}
```

Can't change the memory bubble though..

```
public void foo2() {  
    int[] array = new int[5];  
    fill2(array);  
    for(int i=0; i<array.length; i++)  
        System.out.println(array[i]);  
}
```

```
public void fill2(int[] theArray) {  
    theArray = new int[5];  
    for(int i=0; i<theArray.length; i++)  
        theArray[i] = i;  
}
```