Arrays can be confusing. Students sometimes get to arrays, and go no further. This has been true for all programming languages---BASIC, FORTRAN, Pascal, and Java. The problem with arrays is that statements must now do indexing in addition to doing their usual work. Assignment statments used to be simple:

```
sum = result + value;
```

Now they can be complicated:

```
sum = result[j] + value[k];
```

Actually, arrays are not difficult once you get accustomed to the notation, and that just takes some practice. These exercises provide practice in using subscripts.

Copy each skeleton program to your programming editor and complete it there as directed. Then compile and run the program.

Exercise 1 --- Array Sum

Examine the following program:

```
class Exercise1
{
  public static void main ( String[] args )
  {
    int[] val = {0, 1, 2, 3};

    sum =

    System.out.println( "Sum of all numbers = " + sum );

   }
}
```

Complete the assignment statement so that it computes the sum of the numbers in the array.

Click here to go back to the main menu.

Exercise 2 --- Two Arrays

Examine the following program:

```
class Exercise2
{
```

```
  public static void main ( String[] args )
  {
    int[] val = {13, -4, 82, 17};
    int[] twice;

    System.out.println( "Original Array: "
        + val[0] + " " + val[1] + " " + val[2] + " " + val[3] );

    // Construct an array object for twice.


    // Put values in twice that are twice the
    // corresponding values in val.



    System.out.println( "New Array: "
        + twice[0] + " " + twice[1] + " " + twice[2] + " " + twice[3] );
   }
}
```

Complete the program so that a new array `twice` is constructed. Now copy values from `val` to `twice`, but make the values in `twice` double what they are in `val`.

Click here to go back to the main menu.

Exercise 3 --- Three Arrays

Examine the following program:

```
class Exercise3
{
  public static void main ( String[] args )
  {
    int[] valA   = { 13, -22,  82,  17};
    int[] valB   = {-12,  24, -79, -13};
    int[] sum    = {  0,   0,   0,   0};

    // Add values from corresponding slots of valA and valB
    // and put the result in the corresponding slot of sum.



    System.out.println( "sum: "
        + sum[0] + " " + sum[1] + " " + sum[2] + " " + sum[3] );
   }
}
```

Complete the program with four assignment statements so that each slot of `sum` contains the sum of the corresponding slots in `valA` and `valb`. Ie., add slot zero of `valA` to slot zero of `valB` and put the result in slot zero of `sum`, and so on.

Click here to go back to the main menu.

Exercise 4 --- Same Sum

Examine the following program:

```
class Exercise4
{
  public static void main ( String[] args )
  {
    int[] valA   = { 13, -22,  82,  17};
    int[] valB   = {  0,   0,   0,   0};

    // Put values into valB so that the sum of the values
    // in corresponding slots of valA and valB is 25.




    System.out.println( "valA: "
        + valA[0] + " " + valA[1] + " " + valA[2] + " " + valA[3] );

    System.out.println( "valB: "
        + valB[0] + " " + valB[1] + " " + valB[2] + " " + valB[3] );

    System.out.println( "sum:  "
        + (valA[0]+valB[0]) + " " + (valA[1]+valB[1]) + " "
        + (valA[2]+valB[2]) + " " + (valA[3]+valB[3]) );
   }
}
```

Complete the program with four assignment statements that put values into `valB` so that the sum of corresponding slots in `valA` and `valB` is 25.

Click here to go back to the main menu.

Exercise 5 --- Reverse Order

Examine the following program:

```
class Exercise5
```

```
{
  public static void main ( String[] args )
  {
    int[] val = {0, 1, 2, 3};
    int temp;

    System.out.println( "Original Array: "
        + val[0] + " " + val[1] + " " + val[2] + " " + val[3] );

    // reverse the order of the numbers in the array



    System.out.println( "Reversed Array: "
        + val[0] + " " + val[1] + " " + val[2] + " " + val[3] );
   }
}
```

Complete the program so that the numbers in the array appear in reversed order. You will need to use the variable `temp` to do this.

Note: this is a harder exercise than you might guess. I sometimes put it on a midterm examination, and some students get the question wrong.

Click here to go back to the main menu.