Arrays

Let's take a really simple example; the array will be declared for its exact size, then we'll traverse the array forwards and backwards:

```
import components.simplereader.SimpleReader;
import components.simplereader.SimpleReader1L;
import components.simplewriter.SimpleWriter;
import components.simplewriter.SimpleWriter1L;

/**
 * Demo of arrays.
 *
 * @author Chris Kiel
 *
 */
public final class ArrayDemo {

    /**
     * Private constructor so this utility class cannot be instantiated.
     */
    private ArrayDemo() {
    }

    /**
     * Main method.
     *
     * @param args
     *            the command line arguments
     */
    public static void main(String[] args) {
        SimpleReader in = new SimpleReader1L();
        SimpleWriter out = new SimpleWriter1L();
        final int MAX_SIZE = 5;

        int i;
        int nums[] = new int[MAX_SIZE]; //array declaration, allocation

        for (i = 0; i < nums.length; i++) {
            nums[i] = i * i;
        }

        out.println("Output in original order: ");
        for (i = 0; i < nums.length; i++) {
            out.print(nums[i] + "  ");
        }
        out.println();

        out.println("Reverse of original order: ");
        for (i = nums.length - 1; i >= 0; i--) {
            out.print(nums[i] + "  ");
```

```
        }
        out.println();

        /*
         * Close input and output streams
         */
        in.close();
        out.close();
    }

}

/*
 * Output in original order:
 * 0 1 4 9 16
 * Reverse of original order:
 * 16 9 4 1 0
 */
```

Sometimes, you don't know ahead of time how much data you will need to store in an array. The typical solution then is to declare an array that will be big enough, knowing that there will be some wasted space.

```
import components.simplereader.SimpleReader;
import components.simplereader.SimpleReader1L;
import components.simplewriter.SimpleWriter;
import components.simplewriter.SimpleWriter1L;

/**
 * Demo of arrays where you don't know how many elements you will have.
 *
 * @author Chris Kiel
 *
 */
public final class BigEnoughArrayDemo {

    /**
     * Private constructor so this utility class cannot be instantiated.
     */
    private BigEnoughArrayDemo() {
    }

    /**
     * Main method.
     *
     * @param args
     *            the command line arguments
     */
    public static void main(String[] args) {
        SimpleReader in = new SimpleReader1L();
```

```
        SimpleWriter out = new SimpleWriter1L();
        final int MAX_SIZE = 50;

        int i, number, n = 0;
        int nums[] = new int[MAX_SIZE]; //array declaration, allocation

        out.print("Enter an integer, -1 to quit: ");
        number = in.nextInteger();
        while (number != -1) {
            nums[n] = number;
            n++;
            out.print("Enter an integer, -1 to quit: ");
            number = in.nextInteger();
        }

        out.println("\nThere were " + n + " numbers.");
        out.println("Output in original order: ");
        for (i = 0; i < n; i++) {
            out.print(nums[i] + "  ");
        }
        out.println();

        out.println("Reverse of original order: ");
        for (i = n - 1; i >= 0; i--) {
            out.print(nums[i] + "  ");
        }
        out.println();

        /*
         * Close input and output streams
         */
        in.close();
        out.close();
    }

}
/*
 * Enter an integer, -1 to quit: 56
 * Enter an integer, -1 to quit: 85
 * Enter an integer, -1 to quit: 42
 * Enter an integer, -1 to quit: 21
 * Enter an integer, -1 to quit: 97
 * Enter an integer, -1 to quit: 56
 * Enter an integer, -1 to quit: -1
 *
 * There were 6 numbers.
 * Output in original order:
 * 56 85 42 21 97 56
 * Reverse of original order:
 * 56 97 21 42 85 56
 */
```

Are Arrays Objects?

- Arrays are (mostly) treated as objects:
  - Instantiated with the **new** operator.
  - Have instance variables (e.g., length).
  - Array variables are *reference variables*.
  - As a parameter, a reference to the array is passed rather than copies of the array's elements.
- But…
  - There is no Array class. So arrays don't fit into the Object hierarchy.