Please read Sections 7.1 through 7.6 of the textbook and then answer the following, trying not to look at your notes or at the textbook. Quiz #9, on Fri. 4 Nov., will consist exclusively of questions taken from the Part 1 of this homework.

Ex. 1. Can an array hold values of different data type (for instance, `char` and `double`)?

Ex. 2. Write a statement that creates a 10-element `int` array referenced by the variable `numbers`.

Ex. 3. In the following, what is the value of the size declarator? What is the value of the subscript?

```
int[] numbers;
numbers = new int[8];
numbers[4] = 9;
```

Ex. 4. What is wrong with the following array declaration?

```
int[] books = new int[-1];
```

Ex. 5. Draw the content of the `scores` array once those statements have been executed

```
int[] scores = new int[3];
scores[0] = 13;
scores[2] = 25;
```

Ex. 6. What would be printed by the following code?

```
int[] values = new int[6];
for (int i = 0 ; i < 6 ; i++)
    values[i] = (i*2);
for (int j = 0 ; j < 6 ; j++)
    System.out.println(values[j]);
```

Ex. 7. Suppose the variable `dailyPushUp` references an `int` array with 7 elements. Write a `for` loop that prints each element of the array `dailyPushUp`.

Ex. 8. What is "array bounds checking"? When does it happen?

Ex. 9. Is there an error with the following code? If you think there is one, explain it, otherwise draw the content of the `myIncomes` array once those statements have been executed.

```
double[] myIncomes = new double[5];
myIncomes[1] = 3.5;
// No income on day two.
myIncomes[3] = 5.8;
myIncomes[4] = 0.5;
myIncomes[5] = 1.5;
```

Ex. 10. What would be the size of the `test` array after the following statement has been executed?

```
int[] test = {3, 5, 7, 9};
```

Ex. 11. What is the difference, if any, between the following two statements?

```
int[] num, scores;
int num[], scores;
```

Ex. 12. Write a statement that creates and initializes a `double` array with the values 12.5, 89.0 and 3.24.

Ex. 13. What is the value of `count` and the content of `number` once the following has been executed?

```
int count=2;
int[] number={3, 5, 7};
number[count--] = 8;
number[count]--;
```

Ex. 14. Suppose we have an array named `temp` that has been declared and initialized. How can we know the number of elements in this array?

Ex. 15. Describe what the following code would do.

```
int[] record={3, 8, 11};
int accumulator = 0;
for (int i : record)
accumulator += i;
```

Ex. 16. Assuming we have two `int` arrays of the same size, `firstA` and `secondA`, write a statement that copy the content of `firstA` into `secondA`.

Ex. 17. Write a method (header included) that takes as argument an `int` array, and prints the value of each element of that array.

Ex. 18. Write a method (header included) that takes as argument an `int` array, and stores the value 10 in each element of that array.

Ex. 19. What is the return type of a method returning an `String` array?

Ex. 20. Suppose `listOfGuests` is a `String` array. Write a statement that prints the length of the third element in `listOfGuests`.

Part II — Programming Exercises

Ex. 1. We're going to write a "library" of methods to manipulate `double` arrays. More precisely, we want to write the following (`static`) methods:

- A `printArray` method, that takes as argument a `double` array, and prints the content of each of its element.
- A `copyArray` method, that takes as argument two `double` arrays, and copy the content of the first one in the second one *if the two arrays have the same size*.
- A `setArray` method, that takes two arguments: a `double` array, and a `double`. This method should set the value of the each element of the array to be the `double` given as argument.
- An `avgArray` method, that takes a `double` array as argument, and return the average of all the values in that array.

Start by thinking about your program, and if you feel like you're completely stuck, or to make sure that your code is sensible, you can compare what you wrote with section 7.4 of your textbook. Use the following application program to make sure that your methods are sensible:

```
public class ArrayLib{
  public static void main(String[] args){
    double[] test1 = {3.5, 4, 4.5};
    double[] test2 = new double[3];
    printArray(test1);
    // Should print 3.5 4.0 4.5
    copyArray(test1, test2);
    test2[0]+=3;
    printArray(test2);
    // Should print 6.5 4.0 4.5
    setArray(test1, 3);
    System.out.print(avgArray(test1));
    // Should print 3.0
    System.out.print(avgArray(test2));
    // Should print 5.0
  }
}
```

Of course, you're welcome to adapt this application program to make more tests!