CSC 205 Review Homework for Course Portfolio

1. Give the declarations needed to do each of the following:
   a. A 24-element `double` array

```
double[][] number = new double[23][23];
```

   b. A 500-element `int` array

```
int[] number = new int[499];
      for (int i = 0; i < 499; i++)
            number[i] = (int)(Math.random()*100) + 1;
System.out.println(Arrays.toString(number));
```

   c. A 50-element array of `String` objects

```
String[] objects = new String [50];
      // objects[0] = ;
      // ...
      // objects[49] = ;
System.out.print(Arrays.toString(objects));
```

   d. A 10-element `char` array

```
char[] character = new char[] {'a','b','c','d','e','f','g','h','i','j'};
    String print = new String(character);
System.out.println(print);
```

2. Given the declarations:

```
      final int SIZE = 10;
      int[] count = new int[SIZE];
      String[] rainbow = {"BLUE", "RED", "GREEN", "RED", "GREEN"};
```

Write code fragments (not a complete method) to do the following tasks:
   a. Set `count` to all zeroes

```
int count = 0;

// answer
for (int i = 0; i < count.length;  i++)
     count[i] = 0;
```

   b. Change the second element in `rainbow` to "WHITE"

```
rainbow[1] = "WHITE";
System.out.print(rainbow[1]);
```

   c. Count the number of times "GREEN" appears in `rainbow`

```
for (int i = 0; i < rainbow.length; i++)
      if (rainbow[i].equals("GREEN"))
            count++;
System.out.print(count);
```

   d. Print out all the strings in `rainbow` on separate lines

```
System.out.println(rainbow[0]);
System.out.println(rainbow[1]);
System.out.println(rainbow[2]);
System.out.println(rainbow[3]);
System.out.println(rainbow[4]);
```

e. Sum the values in `count`

```
for (int i = 0; i < rainbow.length; i++)
      if (rainbow[i].equals(rainbow))
            count++;
System.out.print(count);
```

3. Write a code segment (not a complete method) to do each of the following:

a. Declare a two-dimensional `int` array named `table` with 4 rows and 3 columns.

```
int[][] table = { {64, 3, 11,},
                  { 8, 7,  9,},
                  { 1, 1, 45,},
                  {34, 2,  6 }
                };
```

b. Find the sum of the `table` array. Assume it has already been initialized.

```
191

// answer
int sum = 0;
for (int row = 0;  row < table.length;  row++)
    for (int col = 0; col < table[0].length;  col++)
                  sum += table[row][col];
```

4. Given the following declarations:

```
      final int NUM_STUDENTS = 100;
      boolean[] failing = new boolean[NUM_STUDENTS];
```

Write a Java class method that initialized all the components of the `failing` array to `false`. Think about whether your method should be `void` or `value-returning` and use the appropriate one. Give a sample call to your method.

```
      private static void initialize (boolean[] failing) {
            for (int i = 0; i < failing.length; i++)
                  failing[i] = false;
            }

// answer
/* Notice the name static in the method header.  That makes it a class
method. You are sending it one parameter, an array of booleans.  It is a
void method since it does not return anything.  All it does is initialise
the array. Array parameters are pass by reference and preserve changes made
within the method so, after we assign false to all, those changes stay
locked in. */
```