

Assignment 07 – Java Concepts Chapter 07: Arrays & Array Lists
Exercise P7.9

Due: Monday, Nov 10th

Write a program that produces random permutations of the numbers 1 to 10. To generate a random permutation, you need to fill an array with the numbers 1 to 10 so that no two entries of the array have the same contents. You could do it by brute force, by calling `Random.nextInt` until it produces a value that is not yet in the array. Instead, you should implement a smart method. Make a second array and fill it with the numbers 1 to 10. Then pick one of those at random, remove it, and append it to the permutation array. Repeat 10 times. Implement a class `RandomPermutationGenerator` with a method `int[] nextPermutation`.

Note: `Random.nextInt` is not a static method – in order to use it, you'll need to build a new object of the `Random` class first & then call the method using that object. For instance:

```
// build a new random number generator
Random randGen = new Random();
// get a random number between 0 & 9
int number = randGen.nextInt(10);
```

See the page below for some starting code. As before, be sure to include documentation commenting and pay close attention to any loops in your code. Include a couple of comments around each.

Extra credit: Write an *additional* new class called `ListPermutationGenerator`. It should work the same way as your `RandomPermutationGenerator` except it should use an `ArrayList` of integers instead of an array of integers. You'll need to supply an additional tester as well.

Extra credit: Look through the Java API for a method called `shuffle`. Write an *additional* new class called `EasyPermutationGenerator` that with a method `int[] nextPermutation` that uses `shuffle` to build a random permutation of an integer array.

Put a copy of your source code files, into a **.zip** file named **CS125-A07-YOURNAME.zip**. The only thing in the zip should be your source files, & possibly some folders to organize them. Note: if you have trouble creating zip files in the labs, refer to my tutorial on my CS104 course page.

Upload the **.zip** file to Moodle.

The following rubric will be used for grading:

Description	Points
Correct filename(s) are used	1
Source code content – classes implemented as indicated in text	3
Source code compiles without errors	3
Program executes	2
Program output is correct	4
<code>PermutationGenerator</code> uses <code>Random.nextInt</code>	1
Any/all classes are documented	3
Any/all methods & constructors are documented	4
Any/all parameters and return values are documented	2
Loops are briefly commented	2
TOTAL POSSIBLE POINTS:	25
Extra credit, working, documented <code>ListPermutationGenerator</code>	5
Extra credit, working, documented <code>EasyPermutationGenerator</code>	5

Use the following class as your main class:

```
/**
 * This class prints 5 permutations of the numbers 1 through 10.
 */
public class PermutationPrinter
{
    public static void main(String[] args)
    {
        RandomPermutationGenerator gen = new RandomPermutationGenerator (10);
        for (int i = 1; i <= 5; i++)
        {
            for (int n : gen.nextPermutation())
                System.out.print(" " + n);
            System.out.println();
        }
    }
}
```

Use the following class in your solution:

```
import java.util.Random;

/**
 * This class generates random permutations of a sequence of integers
 * 1...length.
 */
public class RandomPermutationGenerator
{
    /**
     * Constructs a RandomPermutationGenerator object.
     * @param length the length of the permutations generated
     * by this generator.
     */
    public RandomPermutationGenerator(int length)
    {
        // fill this in
    }

    /**
     * Gets the next permutation.
     * @return the array containing the next permutation
     */
    public int[] nextPermutation()
    {
        // fill this in
    }

    // you'll need at least one instance field to keep track
    // of the length passed into the constructor
    // you'll need to decide if you need more than that though
}
```