

CS526
Homework Assignment 1

Due: 9/16

The goal of this assignment includes:

- Practice of using basic Java features, including arrays, loops, selection statements, and simple I/O.
- Practice of defining a Java subclass.

Part 1. Create a file named *Hw1_part1.java* and write the following methods in it:

- Write a Java method named *sumOfSquaresOfOdds* that receives an integer n and returns the sum of squares of all positive odd integers less than or equal to n .
- Write a Java method named *shuffle* that receives an array of integers and exchanges the first half of the array with the second half of the array. Two examples are shown below:

Example 1: Given array a has an even number of integers:

Initial array: $a = [5, -3, 7, 8, 1, 24]$

After the method is executed: $a = [8, 1, 24, 5, -3, 7]$

Example 2: Given array a has an odd number of integers:

Initial array: $a = [5, -3, 7, 100, 8, 1, 24]$

After the method is executed: $a = [8, 1, 24, 100, 5, -3, 7]$

- Write a Java method named *statistics* that receives an array of double numbers, calculates the largest number, the smallest number, and the average of all numbers. The method, then, stores these three statistics, in that order, in an array of size 3 and returns the array.
- Write a *main* method to test above three methods:
 - Invoke *sumOfSquaresOfOdds* twice once with $n = 10$ and then with $n = 20$, and print each result on the screen.
 - Create an array with 20 integers, invoke *shuffle* and pass this array as an argument, and print the result on the screen. Repeat this with 21 integers.
 - Create an array with 10 double numbers, and invoke *statistics* and pass this array as an argument, and print the result (max, min, average) on the screen. Repeat this with different set of 10 double numbers.

Part 2. **Note:** For this problem, you first need to study the *Progression* class and its subclasses that are described in Section 2.2.3 of the textbook. Create a file named *Hw1Progression.java* that defines the *Hw1Progression* class as a subclass of the *Progression* class. This progression generates a sequence of natural numbers and the next value is determined as follows:

- If the current value x is evenly divisible by 2, the next value is $2x - 1$.
- If the current value x is evenly divisible by 3, the next value is $3x - 2$.
- Otherwise, the next value is $x + 1$.
- Note that if the current value is evenly divisible by both 2 and 3 (such as 6 or 12), then the next value should be $2x - 1$.
- For example,
 If the first value of the sequence is 1 (default value), the following sequence is created:
 1 2 3 7 8 15 43 44 87 259
 If the first value of the sequence is 4, the following sequence is created:
 4 7 8 15 43 44 87 259 260 519

The class definition must include:

- A default constructor that starts with 1.
- A parameterized constructor that starts with a specified natural number.
- An *advance()* method that determines the next value.

Then, modify the *main* method of the *TestProgression.java* file by adding a code segments that performs the following test:

- Create an *Hwl* progression with a default constructor and print the first 10 values on the screen.
- Create an *Hwl* progression with 5 as the start value and print the first 10 values on the screen.

Documentation

No separate documentation is needed. However, you must include sufficient inline comments within your program.

Deliverables

You must submit *Hwl_part1.java*, *HwlProgression.java*, and modified *TestProgression.java* files. Combine all three files (and other additional files, if any) into a single archive file, name it *LastName_FirstName_hw1.EXT*, where *EXT* is an appropriate file extension, such as *zip* or *rar*, and upload it to Blackboard.

Grading

Each method (except the main method) in Part 1 is worth 10 points. Part 2 is worth 10 points. So, total possible points are 20 points.

Each method in Part 1 and the *HwlProgression* class in Part 2 will be tested with different input arguments and points will be deducted, up to 2 points per each method of Part 1 and up to 6 points for the *HwlProgression* class of Part 2, if your output is incorrect.

Up to 4 points will be deducted if your program does not have sufficient inline comments.