Lab 9: Classes, Methods, Arrays I
20 pts

Distribute on December 1, 2025
Due December 7, 2025 (Sunday) at 11:59 pm before 12:00 midnight

Learning Outcomes (CLO vs SO Mapping)

- Utilize Java syntax in fundamental programming algorithms (3)vs(1)
- Recognize and apply the various control structures (5)vs(1)
- Design and implement elementary multi-class solutions to programming problems (6)vs(2, 6)
- Recognize and apply the basic debugging strategies in programming (8)vs(2)
- Recognize the need for arrays in the solutions of programming problems, and manipulate data in one-dimensional arrays (7)vs(1, 6)

Requirements

This lab is primarily designed to provide you with hands-on experience in working with arrays, array manipulations, copy constructors, and multi-class design. The ArrayPractice class, being a key component of this lab, will be the main focus of your exercise.

Preliminaries

1. Create a Java project; the name of the project must be **<your-pfw-id>_lab9**. For example, my project would be named pNg_lab9.

2. Add three Java classes to the project: **Applications** (contains the main method), **Rectangle** (copy over here the Rectangle class you implemented in Lab 8), and ArrayPractice

3. Add the following comment block to the beginning of each of your classes:

```
/*
 * <your name>
 * CS 16000-01 02/03, Fall Semester 2025
 *  (Note: Write either 02 or 03, depending on which section your section is.)
 * Lab 9
 */
```

4. Make additional documentation about the purpose of the classes and methods to provide clear guidance on their functionality and usage to provide clear guidance on their functionality and usage.

5. **To ensure you receive credit for your work, it is crucial that the TA verifies your attendance in the lab.**

***Exercising the* ArrayPractice** class

This exercise aims to practice declaring, instantiating, and manipulating array variables using the ArrayPractice class, a key and engaging component of this lab.

1. (1pt) The **ArrayPractice** class has three private data fields, all arrays are arrays, and an additional constant field **baseLength** is assigned 10. Declare the array fields as
   - **a)** an array to store integer numbers of type int named **numbers**
   - **b)** an array to store Rectangle references (type is Rectangle) named **boxes**
   - **c)** an array to store names (type String), the id is **listOfNames**

   Do not instantiate the fields in the declaration!

2. (2pts) Define a constructor for this class such that it takes no parameters, and the constructor instantiates all array fields to length baseLength.

3. (3pts) Add another constructor to this class such that it takes three parameters,
   a) an int for the length of the numbers array,
   b) another int for the length of the boxes array, and
   c) a String type array to initialize the listOfNames array.

   The constructor instantiates numbers and boxes to the given parameters as lengths and assigns the third parameter to the String array. Choose your own IDs for the parameters. The constructor calls the loadNumbers() and loadBoxes() methods (see below).

4. (3pts) Define the loadNumbers( ) method. The task of this method is to populate the numbers array with randomly selected integer numbers from the range -100 to 100. Determine the return type and parameters as needed. The method must run a for loop to the length of the array and assign a random number one by one to the array entries.

5. (3pts) Define the loadBoxes( ) method. The task of this method is to populate the boxes array with Rectangles. Determine the return type and parameters as needed. The method must run a for loop for the length of the array. For every index k, a Rectangle object is instantiated with randomly selected length and

width values between 0 and 1. Use the initializer constructor from the Rectangle class and assign the instantiated rectangle to boxes[k].

6. (3pts) Define three more methods: displayNumbers, displayBoxes, and displayList. The task of each method is to print the corresponding array entries to the console, each array entry on a new line. A rectangle should be printed using the toString method, defined in the Rectangle class.

7. (5 pts) The Application class contains the main method. In the main()

    1) Declare and instantiate an ArrayPractice object with the no-arg constructor. Call the display methods one after the other. Comment on the output you see on the console.
    2) Using the initializer list, instantiate a String array named **list** containing names of your choice. The array must contain at least 5 names, and it is recommended that it should not exceed 10 names.
    3) Declare and instantiate an ArrayPractice object with the parameterized constructor. For the length of the variable numbers, the parameter is 50; for box length, the parameter is 6; for the list of names, pass the list array you defined above.
    4) Call the display methods again, one after the other. Comment on the output you see on the console.

Submit

- Zip your project folder and submit your assignment on Brightspace at the designated link