

CPSC 1061 – Introduction to Programming in Java Lab

Spring 2021

Lab 11 – Due Monday, April 12, 10:00pm

1 Introduction and Lab Objectives

In this lab you will continue practice using classes and objects in Java programs. The objectives of this lab are to:

1. create and work with classes and objects
2. use some of the existing Java classes, such as BigInteger and Time
3. use a regular expression

2 Main

2.1 General Instructions

At the start of each program, write your name, the course and lab, the date, and a description of what your program does similar to the previous lab. Each program needs to have comments (not just at the beginning), to be clean, and to compile. Furthermore, any input and output should be designed to have appropriate instructions and sentences.

We are writing programs with multiple classes. You can submit different Java classes as different files or within the same file as long as the main program is named correctly (and of course compiles correctly).

2.2 Triangle.java and Perimeter.java (and Point.java) (4 points)

Create a program Perimeter.java that includes the classes Point and Triangle based on the specifications below.

Use your Point class from lab 9. Here is a reminder of the specification:

Point	
-x: double -y: double	The x-coordinate of the point The y-coordinate of the point
+Point(x: double; y: double) +printPoint(): String +distance(p1: Point): double	Creates a point in 2D with an x-coordinate and a y-coordinate Returns the coordinates of a point as a String "(x, y)" that can be printed Computes the distance between two points (the instance point and p1)

As previously, create three instances of your Point class a, b, and c. The point a is located at the coordinates (1.2,-0.4), b at (11.7, -0.4), and c at (1.2,13.6).

Next, create and test the class Triangle with its data fields, constructor and methods based on the specification below:

Triangle	
-p1: Point -p2: Point -p3: Point	The first point (vertex) of the triangle The second point (vertex) of the triangle The third point (vertex) of the triangle
+Triangle(p1: Point; p2: Point; p3: Point) +printTriangle(): String +perimeter(): double	Creates a triangle out of 3 points Returns a String of the coordinates of the triangle, e.g., [(1.0, 2.2), (3.5, 4.9), (5.1, 6.2)] Returns the perimeter of the triangle

In your main method in the class Perimeter.java, create a triangle (an object of the class Triangle) that is defined by the points a, b, and c. Display (print) the coordinates of the points of the triangle. Compute and display the perimeter of the triangle. Reuse the methods of the classes when possible: use the distance method in the perimeter method, use the printPoint method in the printTriangle method.

2.3 ApplesAndBananas.java

Start with the String

I like to eat, eat, eat apples and bananas.

Use a regular expression to elegantly replace each vowel (a, e, i, o, u) or set of two vowels with *ay*. The result should be *ay laykay tay ayt, ayt, ayt aypplays aynd baynaynays*.

Print both sentences.

2.4 AroundTheWorld.java (4 points)

In this part of the lab, we are going to learn how to read content from a file using the File and Scanner classes.

Use the documentation to explore these classes as needed:

<https://docs.oracle.com/en/java/javase/15/docs/api/java.base/java/util/Scanner.html>

<https://docs.oracle.com/en/java/javase/12/docs/api/java.base/java/io/File.html>

Follow these steps:

1. Create a java-file AroundTheWorld.java with a main method.
2. Download the file JV.txt to the same folder that AroundTheWorld.java
3. We first need to import the File and the Scanner classes:

```
import java.util.Scanner;
import java.io.File;
```

4. The main method needs to start as follows:

```
public static void main(String[] args) throws Exception
```

We are going to learn more about exceptions in Chapter 12.

5. Create an object of the File class using the constructor of the File class that uses the pathname (see documetation). The pathname is

JV.txt

6. Continuing in the File class documentation, get familiar with the exists() method. Write an if-statement that uses the exists() method to only continue if the file exists
7. Have a look at the Scanner class documentation. Get familiar with the methods hasNext() and nextLine()
8. Use these methods to read in each line of the file as long as hasNext() returns true
9. Count the number of lines in the file
10. Print the result

The file I used is the full text of the novel “Around the World in 80 Days” by Jules Verne. I downloaded it from Project Gutenberg. To comply with their license, here is a brief statement:

This eBook is for the use of anyone anywhere at no cost and with almost no restrictions whatsoever. You may copy it, give it away or re-use it under the terms of the Project Gutenberg License included with this eBook or online at www.gutenberg.org

The full license is included at the end of the file.

2.5 Submit Files

Make sure to test your java files on the lab machines. Create a single zip-file that includes all the java files (and the txt file) and submit the zip-file to Canvas.