

# CECS 277 LAB I/O EXCEPTIONS

**OBJECTIVE:** Get experience processing ASCII data, handling errors in the input, working with home-grown exception classes.

**INTRODUCTION:** Please remember the coding standards [here](#).

One of the reasons that this course treats exceptions at the same time that we teach you about I/O is because there is a lot that can go wrong when reading/writing files in Java, and the best way to deal with those occurrences is with an exception. In this lab, we will:

1. Prompt the user for an input file until they give us a file that exists.
  - a. You will first off have to create a `File` object from the `String` name of the file.
  - b. Then create a `Scanner` object from the `File` object that you just created.
  - c. If the file does not exist, this will throw a `FileNotFoundException`. You need to catch that exception, alert the user that they blew it, and prompt them again, until they give you a valid path and file name.
2. Prompt the user for an output file until they give us a path and file name that we can write to.
  - a. You will need to create a `PrintWriter` object using the `String` name of the path and file that you want to write out to.
  - b. If the path is invalid, you will get a `FileNotFoundException` here as well. Be sure to trap that exception and prompt the user for the output file again until they get it right.
3. Read through the input file one line at a time and get the 6 double precision numbers from each line. Those six numbers represent the coordinates for three separate points, which make up the vertices of a triangle.
  - a. Use your input `Scanner` to read the next line.
  - b. Create a `Scanner` with the `String` that you just read in.
  - c. Use your `getNextPoint()` function (from the [IOExceptionRunner.java](#) file to read your three points in, one at a time, from that second `Scanner`.
  - d. If you are unable to read in the three points for any given line, trap the exception that `getNextPoint()` throws, and write out a warning to the output file to signify that that input record was corrupt.
4. Then pass your array of three points to your `Area` routine to find the area of the triangle.
5. Write out the area to the output file.
6. The class file for `Point` is [here](#).

## CECS 277 LAB I/O EXCEPTIONS

### PROCEDURE:

1. Create a text file with several legitimate triangles in it. One example is [here](#).
2. Get your application to work with that simple input.
3. Then, try removing one of the numbers from one of the records in the file to create a corrupt row and make sure that it works.
4. Then, try putting in something in one of the records that is not a double precision number and make sure that you get the proper results from that.
5. In your final run, have a mixture of records that are good, that are missing one or more numbers, and records that have non-numeric data.

### WHAT TO TURN IN:

- Your updated IOExceptionRunner.java.
- Your input file.
- Your output file.