Programming Project #1

CIS 296 – Fall 2018 – Prof. John P. Baugh
University of Michigan - Dearborn

Points: **_____ / 100**
Due: **October 12, 2018 at 11:59 p.m.**

Objectives

- To solve problems using `ArrayLists`
- To perform file I/O
- To apply geometric concepts, translated into Java

Introduction

This program will allow a user to read the data from file about **spheroids**. Spheroids are important forms in various applications of geometry, such as optometry. The eye, of course, is a spheroid, and depending on the extremes of shaping, different conditions of the eye can be diagnosed.

Instructions and Information

Background Information

You will read the data, store it appropriately in an object of type **Spheroid**, and produce the volume and whether the sphere is oblate or prolate. Obviously, you must create a custom class (that **must be named Spheroid**), and you must provide a means to store the required data.

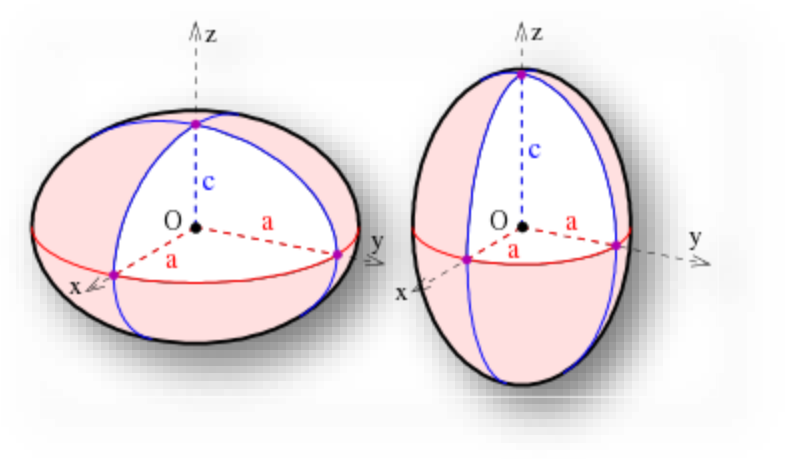

FIGURE 1: OBLATE (LEFT) VS PROLATE (RIGHT) SPHEROIDS[1]

The required data consist of the **equatorial radius** (**a** in Figure 1 above) and the **polar radius** (**c** in the same figure above.)

Some information related to spheroids:

- A spheroid is **oblate** if c < a
- A spheroid is **prolate** if c > a
- A spheroid is a **sphere** if c = a

The **volume of a spheroid**, regardless of its classification is:

$$V = \frac{4}{3}\pi a^2 c$$

Again, *a* is the equatorial radius, and *c* is the polar radius.

For a sphere, since *a* = *c*, this would reduce to the familiar form $V = \frac{4}{3}\pi a^3$.

Input

The input will be obtained **from a file**, named **input.txt**.

A single line, therefore takes the general form:

`Equatorial_Radius Polar_Radius`

Specifically, a single line of example input might be:

[1] https://en.wikipedia.org/wiki/Spheroid

5 2

Output

As soon as the file is read into memory and the data is placed in the appropriate data structure(s), the following will be printed **to standard output (the console)**, in general form:

```
Volume  Classification
```

Where volume is the calculated volume from the formula, and the classification is **oblate, prolate,** or **sphere**.

Example

Sample Input File

```
5 2
10 10
4 3
6 8
```

Sample Output

```
209.44 Oblate
4188.79 Sphere
201.06 Oblate
1206.37 Prolate
```

Hints and Notable Requirements

You should consider **creating two classes** to help you solve the problem:

- `Spheroid`
  - Represents a single spheroid
  - Maintains the equatorial and polar radii as fields
  - Has a constructor allowing the user to set the equatorial and polar radii (as doubles)
  - Has accessors (getters) and mutators (setters) for both the equatorial radius and polar radius
  - Can return the volume using a method **getVolume**
    - Returns a double
    - Takes no parameters
  - Can return the classification of a spheroid, using a method **getClassification**
    - Returns a string
    - Takes no parameters
    - Calculates the information based on the current equatorial and polar radii
  - **There should be NO print statements inside this class**
- `SpheroidDemo`
  - This class will contain the main method
  - You must read the file
  - You must create instances of Spheroid, and store them in an ArrayList<Spheroid>
    - You can use the constructor of the Spheroid immediately with the equatorial and polar radii that are read from file
  - Once **all** information is read from the file and all the spheroid objects are created, loop through the ArrayList of Spheroids and print out their volumes and classifications to the console

Deliverables

To turn in the assignment, please upload a zip file of a folder containing the **.java files** necessary for the program to run. Upload them to the appropriate assignment directory on Canvas.