Lab 4 – CS151

Create a **multi-class application** that creates geometric entities, and has the ability to calculate their respective volumes and to provide a textual description of each object based on instance variables which can be set at creation, or altered afterward. In addition, each variable type must be able to be read individually. Finally, the program should be able to provide the total volume of capacity in all objects.

Object Type	Parameter(s)	Volume Formula
Sphere	Radius (r)	$\frac{4}{3}\pi r^3$
Rectangular Solid	Length, Width, Height (l,w,h)	$l \times w \times h$
Square-based Pyramid	Side, Height (s,h)	$\frac{1}{3}s^2h$
Ellipsoid	Primary Axis, Secondary Axis, Tertiary Axis (a,b,c)	$\frac{4}{3}\pi abc$

Sequence of Operations

1. Create objects with the following parameters:
   - Sphere of radius 3.25
   - Rectangular Solid of dimensions (4.5, 3.5, 6.0)
   - Pyramid of side 4.5 and height 9.2
   - Ellipsoid of axes (4.5, 3.5, 6.0)

2. Now, print each object (using its toString() method); then, print the total volume of all objects. The output of the toString() method for a sphere might look like:
   "The volume of this sphere is nnn.nnnnnnn" (where nnn.nnnnnnn represents some **double**)

3. After that, modify the objects as follows:
   - Change the radius of the sphere to 3.75
   - Change the height of the pyramid to 8.7
   - Set the secondary axis of the ellipsoid to be 3.2 times the value it currently holds (without manually calculating this – use an accessor to get the current value, and a mutator to set the new value based on it; ask your instructor if this is confusing.)

4. Next, print out the value of the *secondary* axis of the ellipsoid so that it appears as follows:
   "The new value for the ellipsoid's secondary axis is: " followed by that new value.

5. Finally, print the new total volume.

Hints

- The value π is represented by the constant PI from the Math class (i.e. Math.PI), which is a **double**
- Remember that exponentiation (n^x) is represented by Math.pow(n,x), and returns a **double** value
- Finally, don't forget that (4/3) = 1 in Java.