EE2510 - Lab 2: Using Classes

2 weeks total

Goals:

1. Creating and using classes and objects
2. File I/O

Assignment Description:

Overview:

Create a program to read a random set of data from a file and print the data by "type" into a second file

Interface:

The program will ask the user for the input file name and output file name

The dimensions for the objects will be provided in the file on the website along with the file format

The program should read the file until an end of file marker is found (do not hardcode the number or order of objects)

The program should print out the results from each of the getter and calc member functions for each object with appropriate formatting

Structural requirements:

Create 3 geometric classes: Box, Sphere, Square Base Pyramid

The top-level program file (includes main) should be used for control only

You must use the UML described classes

Additional functions may be needed or desired

NO global variables

Grading:

Functionality

Comments – readability

Cleanliness (beauty) of the code

Structure

Documentation

On-time

Deliverables:

All code

Eclipse "project explorer" capture showing all files in the project

Screen capture of program run

Copy of the output file

Hardcopy – no need to put into a PowerPoint or pdf, just print/label/staple

Due: 5:00 pm 1 day after week 4 lab – in the box outside my office

Hints:

You can create an array of objects (you may assume < 20 objects of each type)

```
        Box * boxes = new Box[20];          // Objects
```

and use the array index to create separate objects

Be sure to clean up your dynamic memory before exiting

UML

Box

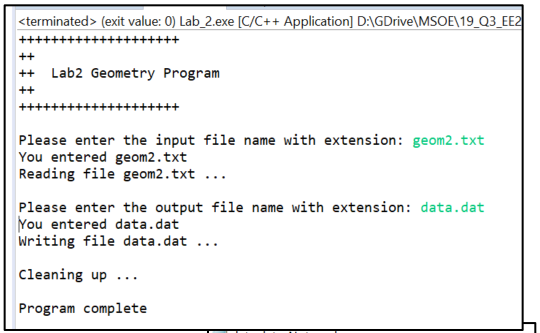
```
- Identity : int
- width : double
- length : double
- depth: double
```

```
+ Box()
+ Box(id : int, w : double, l : double, d : double)
+ ~Box
+ setIdentity(id : int) : void
+ setWidth(w : double) : void
+ setLength(l : double) : void
+ setDepth(d : double) : void
+ getIdentity() : int
+ getWidth() : double
+ getLength() : double
+ getDepth() : double
+ calcVolume() : double
+ calcSurfaceArea() : double
```

Sphere

```
- Identity : int
- radius : double
```

```
+ Sphere()
+ Sphere(id : int, r : double)
+ ~Sphere
+ setIdentity(id : int) : void
+ setRadius(r : double) : void
+ getIdentity() : int
+ getRadius() : double
+ calcVolume() : double
+ calcSurfaceArea() : double
```



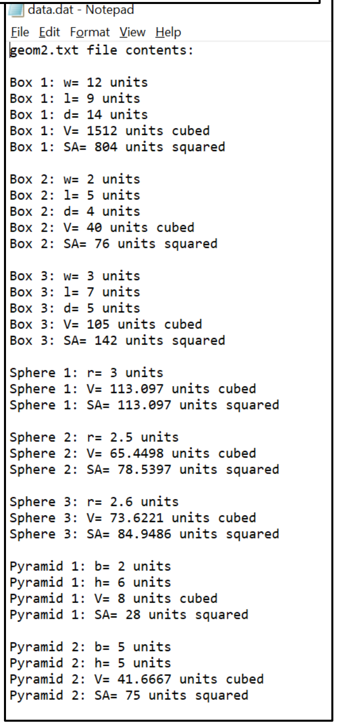
```
<terminated> (exit value: 0) Lab_2.exe [C/C++ Application] D:\GDrive\MSOE\19_Q3_EE2
++++++++++++++++++++
++
++  Lab2 Geometry Program
++
++++++++++++++++++++

Please enter the input file name with extension: geom2.txt
You entered geom2.txt
Reading file geom2.txt ...

Please enter the output file name with extension: data.dat
You entered data.dat
Writing file data.dat ...

Cleaning up ...

Program complete
```



```
geom2.txt file contents:

Box 1: w= 12 units
Box 1: l= 9 units
Box 1: d= 14 units
Box 1: V= 1512 units cubed
Box 1: SA= 804 units squared

Box 2: w= 2 units
Box 2: l= 5 units
Box 2: d= 4 units
Box 2: V= 40 units cubed
Box 2: SA= 76 units squared

Box 3: w= 3 units
Box 3: l= 7 units
Box 3: d= 5 units
Box 3: V= 105 units cubed
Box 3: SA= 142 units squared

Sphere 1: r= 3 units
Sphere 1: V= 113.097 units cubed
Sphere 1: SA= 113.097 units squared

Sphere 2: r= 2.5 units
Sphere 2: V= 65.4498 units cubed
Sphere 2: SA= 78.5397 units squared

Sphere 3: r= 2.6 units
Sphere 3: V= 73.6221 units cubed
Sphere 3: SA= 84.9486 units squared

Pyramid 1: b= 2 units
Pyramid 1: h= 6 units
Pyramid 1: V= 8 units cubed
Pyramid 1: SA= 28 units squared

Pyramid 2: b= 5 units
Pyramid 2: h= 5 units
Pyramid 2: V= 41.6667 units cubed
Pyramid 2: SA= 75 units squared
```

Pyramid

```
- Identity : int
- base : double
- height : double
```

```
+ Pyramid()
+ Pyramid(id : int, b : double, h: double)
+ ~Pyramid
+ setIdentity(id : int) : void
+ setBase(b : double) : void
+ setHeight(h : double) : void
+ getIdentity() : int
+ getBase() : double
+ getHeight() : double
+ calcVolume() : double
+ calcSurfaceArea() : double
```