5. PROGRAMMING ASSIGNMENT 5: UPDATED 8/20/17

Read Deitel: Chapter 9-10, Liang: Chapter 10, 11, 13

Programming: Name your 6 files for PA#5:
- P5.java
- P5Polygon.java
- P5RegHexagon.java
- P5RegHexPrism.java
- P5Rectangle.java
- P5Cuboid.java

DUE: Wednesday, August 30, 2017 @ 6am

You will write a program to calculate the area of a P5RegHexagon, surface area of a P5RegHexPrism and volume of a P5RegHexPrism, area of a P5Rectangle, surface area of a P5Cuboid and volume of a P5Cuboid, using classes and an inheritance hierarchy.

PA#5 consists of 6 public classes (in six separate files: P5.java P5Polygon.java P5RegHexagon.java P5RegHexPrism.java P5Rectangle.java, P5Cuboid.java). In order for the grader program to collect your entire PA#5 source files, these file names MUST be named exactly as specified.

P5Polygon: Abstract superclass of P5RegHexagon and P5Rectangle
P5RegHexagon: Subclass of P5Polygon
P5RegHexPrism: Subclass of P5RegHexagon
P5Rectangle: Subclass of P5Polygon
P5Cuboid: Subclass of P5Rectangle
P5: application (driver program)

NOTE: NO error checking of input data is required with corresponding error messages because data is type "double". In production programming, ALL input is error checked. Grader input will include appropriate positive numeric values.

P5Rectangle.java and P5Cuboid.java - refer to "OOP" handout and hints below on P5RegHexagon and P5RegHexPrism. The no argument P5Rectangle constructor initialize height to 1.1. toString() may look like:

\[ \text{"return ("getName()+\"t" + getSide() + " x " + len );"} \]

P5Polygon.java

```java
//
* P5Polygon()
* P5Polygon(double)
* toString()
* getSide()
* getName()
* getDim(double)
* getDim(double, double)
* getDim(double, double, double)
* perimeter()
* area()
* volume()
*/

- No argument constructor, initialize side to one
- No argument constructor, initialize side to parameter
- Format and return class name
- Generates name of this class object and return as a String
- Assign 1 dimension of 2-D polygon (side)
- Assign 2 dimensions of 2-D polygon (side, height)
- Assign 3 dimensions of 3-D polygon (side, height, length)
- Calculate perimeter of a polygon
- Calculate area of a polygon
- Calculate volume of a polygon
```

a) At least one method is abstract (declaration only).
b) Signature of toString() is: public String toString()
c) getName() definition is: "{ return( this.getClass().getName() + "": " )

d) Methods are declared above so they may be accessed through P5Polygon array.
P5RegHexagon.java
/**
* P5RegHexagon() - No argument constructor
* P5RegHexagon(s) - Overloaded constructor, pass the side parameter to P5Polygon(s)
* toString() - Implement toString, format dimensions of a hexagon
* perimeter() - Override and implement super class perimeter
* area() - Implement super class area, calculate area of a hexagon
**/ 
e) All instance variables are protected access level.
f) Write a no argument constructor, initializing to one.
g) Write an overloaded constructor receiving one parameter type double, passing to overloaded P5Polygon constructor taking a double.
h) Write a method to calculate the perimeter which is six times the side length.
i) Write a method to calculate the area which is $\frac{\sqrt{3}}{4} L^2 \times 6 = \frac{3\sqrt{3}}{2} L^2$
j) Write a toString() method to return a type "String" to format dimensions for printing to be called prt() (P5.java). It may look like:
"return (getName() + "\t" + getSide() + "\t");"

P5RegHexPrism.java
/**
* P5RegHexPrism() - No argument constructor, initialize height to one
* P5RegHexPrism(s,h) - Overloaded constructor, initialize height to parameter
* toString() - Override, return a String with formatted dimensions of a hexagonal prism
* setDim(s,h) - Override, assign dimensions of a hexagonal prism
* area() - Override, super class area, calculate area of a hexagonal prism
* volume() - Override - Calculate volume of a hexagonal prism
**/ 
k) All instance variables are protected access level.
l) Write a no argument constructor, initializing to one.
m) Write an overloaded constructor receiving two parameters type double, initializing data to these parameters. Refer to "Object Oriented Programming" handout page 2 "Circle3(int x1, int y1, int r){...}"

n) Override the method to calculate the surface area, of all sides of hexagonal prism which is: 2 times the area of hexagon base plus perimeter times height

o) Write a method to calculate the volume (height * hexagon base area).

p) Override the toString() method to return a type "String" to format dimensions for printing to be called in prt() (P5.java).

q) Override the method (setDim() ) to assign the data members with the parameters.
Note: Perimeter of Hexagonal prism is perimeter of hexagon base

P5.java
/**
* main() - Instantiate P5RegHexagon, P5RegHexPrism, P5Cuboid, P5Rectangle objects, P5Polygon array, initialize variables, read input, reassign to input values repeat program until user types in a word beginning with a single character of ONLY an 'n' or 'N' .
* prt() - Display output
**/ 
r) All instance variables are private access level.
s) As given, in main() instantiate:
- P5RegHexagon object invoking (calling) the no argument constructor.
- P5RegHexPrism object invoking overloaded ctor passing 2.2 for radius
- P5Rectangle object invoking the no argument constructor.
- P5Cuboid object invoking overloaded ctor passing 2.2 for width, 3.3 for length, and 4.4 for height
- P5Polygon array of 4 references. 1st reference points to P5RegHexagon object, 2nd points to P5RegHexPrism object, 3rd points to P5Rectangle object, and 4th points to P5Cuboid object.
- All values shown are initialized from constructors. Call the above object's methods all through the P5Polygon array.
t) In `prt()` call the `toString()`, `area()`, and `volume()` methods. For PA#5 input, format two places of precision for area() and volume() output.

  Note: In Java, if an object is encountered in a String, then the object's `toString()` is called automatically!
  e.g. P5Cuboid b = new P5Cuboid(); System.out.print(b); // Calls P5Cuboid toString();

u) Use `bundleP5` in your PA5 directory, to turnin multiple files.
PA#5 SAMPLE OUTPUT:

P5RegHexagon:  1.0  has an area: 2.60  perimeter: 6.00
P5RegHexPrism:  1.1 x 2.2  has a surface area: 20.81  volume: 6.92
P5Rectangle:  1.0 x 1.1  has an area: 1.10  perimeter: 4.20
P5Cuboid:  2.2 x 3.3 x 4.4 has a surface area: 62.92  volume: 31.94

Enter Hexagon and Rectangle side (width):  1.2
Enter Rectangle length:  2.3
Enter HexPrism and Cuboid (height):  3.4
P5RegHexagon:  1.2  has an area: 3.74  perimeter: 7.20
P5RegHexPrism:  1.2 x 3.4  has a surface area: 31.96  volume: 12.72
P5Rectangle:  1.2 x 2.3  has an area: 2.76  perimeter: 7.00
P5Cuboid:  1.2 x 2.3 x 3.4 has a surface area: 29.32  volume: 9.38
Want to compute areas (y/n)?  x

Enter Hexagon and Rectangle side (width):  5.6
Enter Rectangle length:  6.7
Enter HexPrism and Cuboid (height):  7.8
P5RegHexagon:  5.6  has an area: 81.48  perimeter: 33.60
P5RegHexPrism:  5.6 x 7.8  has a surface area: 425.03  volume: 635.51
P5Rectangle:  5.6 x 6.7  has an area: 37.52  perimeter: 24.60
P5Cuboid:  5.6 x 6.7 x 7.8 has a surface area: 266.92  volume: 292.66
Want to compute area (y/n)?  nada

SAMPLE INPUT is in bold type.

Verify you TURNED-IN your work by re-typing “bundleP5”