You will write a program that allows the user to compute measurements of various geometric shapes using a menu-driven interface. This homework results in ONE program.

About methods:

Now that we are beginning to write longer programs, we will practice dividing our programs into chunks called static methods. This is a critical habit to develop, because building a program out of methods that work together enables faster software development, because functions(methods) can be tested individually, then combined together into larger functions.

When first thinking about a program, try to envision it as a set of components that all fit together like a puzzle, where each component handles a separate, distinct task. Consider making separate tasks into separate static methods. What you need to do

**First, write five methods that compute and return the following:**

1. **Area of a circle.** This method takes one parameter, the radius.

   public static double areaCircle (int radius)

2. **Area of a rectangle.** This method takes two parameters, the length and width.

   public static int areaRectangle(int length, int width)

3. **Area of a square.** This method takes one parameter, the length of a side.

   public static int areaSquare(int side)

However, because a square is just a special type of rectangle (where the sides have the same length), have this function call your area of a rectangle function to compute the answer. In other words, do not directly compute the area of a square inside this function, instead, call your area of a rectangle function with appropriate arguments and return the value that comes back.
This idea, while probably taken a little too far in this case, is normally good programming practice: We are using an already-solved problem to solve a new problem. Here, a square is just a special case of a rectangle, so use the solution for a rectangle we've already written rather than writing a completely new solution.

4. **The surface area of a rectangular prism.** This method takes three parameters, the length, width, and height.

   public static int areaPrism(int length, int width, int height)

   This method takes three parameters, the length, width, and height. Note that the surface area of this shape can be computed as the sum of the areas of the six faces of the prism. Because each face is a rectangle, call your area of a rectangle function to write this function. You will need to call it three times.

   Again, I know I'm taking the idea of reusing functions to the extreme, but I want you to practice calling functions and capturing return values.

5. **The area of a ring, meaning a circle with an interior circle missing,** like this. This function takes two parameters, the inner radius and the outer radius.

   public static double areaRing(int radOuter, int radInner)

   Your area of a ring function should call your area of a circle method twice. If you write these methods correctly, none of them should contain input statements or print statements. Each one interacts with outside methods only through parameters and return values.

**Second, write your main method to do this:**

The user is first asked what kind of shape they want to calculate the area (or surface area) of. The user will type in the name of the shape as a string (e.g., circle, rectangle, square, prism, or ring). Use if-else statements to figure out what shape they wanted, then ask the user to type in the appropriate attributes of the shape (e.g., for circle you would ask for the area, for square you would ask for the length of a side). If the user types in an invalid shape name, then print an appropriate error message.
Display the area of the shape, followed by a message asking the user if they want
to calculate the area of another shape. If they answer yes, then run the program
again. You should use a while loop for this part.
You may assume the user will type in the name of the shape either in all lowercase,
or with a capital first letter (i.e., your program should work correctly for both
"ring" and "Ring"). You do not have to handle any other type of capitalization. If
the user types in an invalid shape name (like "RiNg" or "trapezoid"), your program
should print an appropriate error message.

Testing your program

You should test your program thoroughly to make sure all of your shape functions
work. You may assume the user will never give "bad" input --- the user will always
type in integers for the numbers, and never negative numbers or zero.

Sample interaction

Note that this is only a sample. Your program should work with any order of
shapes the user wants.

What shape do you want? square
What is the side length? 7
The area of the square is 49
Do you want to calculate another area? yes

What shape do you want? rectangle
What is the length? 8
What is the width? 5
The area of the rectangle is 40
Do you want to calculate another area? yes

What shape do you want? prism
What is the length? 6
What is the width? 5
What is the height? 4
The surface area of the prism is 148
Do you want to calculate another area? yes

What shape do you want? Hexagon
I don't know that shape.
Do you want to calculate another area? yes

What shape do you want? square
What is the side length? 5
The area of the square is 25
Do you want to calculate another area? yes

What shape do you want? Ring
What is the outer radius? 6
What is the inner radius? 4
The area of the ring is 62.83185307179586

What shape do you want? Circle
What is the radius? 6
The area of the circle is 113.03999999999999
Do you want to calculate another area? no