WORKED EXAMPLE 4.1

Computing the Volume and Surface Area of a Pyramid

In this Worked Example, we develop a solution to a computational problem.

Problem Statement Suppose that you are helping archaeologists who research Egyptian pyramids. You have taken on the task of writing a method that determines the volume and surface area of a pyramid, given its height and base length.

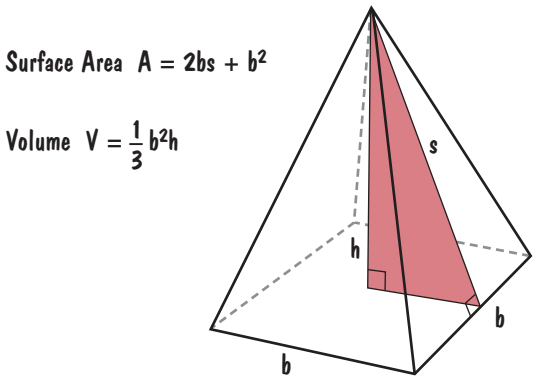
Step 1 Understand the problem: What are the inputs? What are the desired outputs?

Make a list of all the values that can vary. It is common for beginners to implement classes that are overly specific. For example, you may know that the great pyramid of Giza, the largest of the Egyptian pyramids, has a height of 146 meters and a base length of 230 meters. You should *not* use these numbers in your implementation, even if the original problem only asked about the great pyramid. It is just as easy—and far more useful—to write a class that describes *any* pyramid.

In our case, a pyramid is described by its height and base length. The desired outputs are the volume and surface area.

Step 2 Work out examples by hand.

An Internet search yields the following diagram for geometric computations with square-based pyramids:



The volume is straightforward. Consider a pyramid whose base and height are 10 cm each. Then the volume is $1/3 \times 10^2 \times 10 = 333.3$ cm^3, or 1/3 of the volume of a cube with side length of 10 cm. That makes sense if you are familiar with Archimedes' famous decomposition of a cube into three pyramids.

The surface area is not so clear. Looking at the formula $A = 2bs + b^2$, we note that the formula gives the entire area, including the square bottom. That's what you would need if you wanted to find out how much paint you need for a paper model of a pyramid. But do our researchers care about the bottom square that is not exposed? You would need to check back with them. Let's say they reply that they only want the part above the ground. Then the formula becomes $A = 2bs$.

Unfortunately, the value s is not one of our inputs, so we need to compute it. Look at the colored triangle in the figure above. It is a right triangle with sides s, h, and $b / 2$. The Pythagorean theorem tells us that $s^2 = h^2 + (b / 2)^2$.

Now let's try again. If h and b are both 10, then s^2 is $10^2 + 5^2 = 125$, and s is $\sqrt{125}$. Then the area is $A = 2bs = 20 \times \sqrt{125}$, or about 224. This is plausible because four sides of a cube with side

length 10 have area 400, and you would expect that area to be somewhat larger than the four sides of our sample pyramid.

Having solved this example by hand, we are now better prepared to implement the necessary computations in Java.

Step 3 Design a class that carries out your computations.

According to How To 3.1, we need to determine methods and instance variables for our class. In our case, the problem statement yields the following constructor and methods:

- `public Pyramid(double height, double baseLength)`
- `public double getVolume()`
- `public double getSurfaceArea()`

Determining the instance variables requires some thought. Consider these alternatives:

- A pyramid stores its height and base length. The volume and surface area are computed as needed in the `getVolume` and `getSurfaceArea` methods.
- A pyramid stores its volume and surface area. They are computed in the constructor from the `height` and `baseLength`, which are then discarded.

Both approaches will work for our problem. There is no simple rule as to which design is better. One way of settling the question is to consider how the `Pyramid` class might evolve. More methods for geometrical computations (such as angles) might be added. There might be methods to resize the pyramid. The first alternative makes it easier to accommodate those scenarios. Moreover, it seems more object-oriented. A pyramid is described by its height and base, not by its volume and surface area.

Step 4 Write pseudocode for implementing the methods.

As already described, the volume is simply

volume = (base x base x height) / 3

For the surface area, we first need the side length

side length = square root of (height x height + base x base / 4)

Then we have

surface area = 2 x base x side length

Step 5 Implement the class.

As decided in Step 3, we have instance variables for the height and base length:

```
public class Pyramid
{
   private double height;
   private double baseLength;
   . . .
}
```

The methods for computing the volume and surface area are now straightforward.

```
public double getVolume()
{
   return height * baseLength * baseLength / 3;
}

public double getSurfaceArea()
{
   double sideLength = Math.sqrt(height * height
      + baseLength * baseLength / 4);
   return 2 * baseLength * sideLength;
```

```
}
```

There is a minor issue with the constructor. As described in Step 3, the parameter variables of the constructor are identical to those of the instance variables:

```
public Pyramid(double height, double baseLength)
```

One solution is simply to rename the constructor parameter variables:

```
public Pyramid(double aHeight, double aBaseLength)
{
   height = aHeight;
   baseLength = aBaseLength;
}
```

This approach has a small disadvantage. The awkward parameter variable names leak into the API documentation:

```
/**
   Constructs a pyramid with a given height and base length.
   @param aHeight the height
   @param aBaseLength the length of one of the sides of the square base
*/
```

If you prefer, you can avoid that issue by using the `this` reference as follows:

```
public Pyramid(double height, double baseLength)
{
   this.height = height;
   this.baseLength = baseLength;
}
```

We have now completed the class implementation. You can find the complete program in the `ch04/worked_example_1` directory of the book's companion code.

Step 6 Test your class.

We can use the computations from Step 2 as a test case:

```
Pyramid sample = new Pyramid(10, 10);
System.out.println(sample.getVolume());
System.out.println("Expected: 333.33");
System.out.println(sample.getSurfaceArea());
System.out.println("Expected: 224");
```

It is a good idea to have another test case where the height and base are different, to check that the constructor is taking the parameters in the correct order. An Internet search yields an estimate of about 2,500,000 cubic meters for the Giza pyramid.

```
Pyramid gizeh = new Pyramid(146, 230);
System.out.println(gizeh.getVolume());
System.out.println("Expected: 2500000");
```

The program output is:

```
333.3333333333333
Expected: 333.33
223.60679774997897
Expected: 224
2574466.6666666665
Expected: 2500000
```

The answers match well, and we decide that the test was successful.