A Triangle Example

Let's look at an example. To create the `Triangle` (or `Circle` or `Square`) classes, using the abstract Shape class as the base class, all you need do is:

```
class Triangle : public Shape
{
public:
    // MUST override; pure virtual in Shape class
    void draw() const;
};

void Triangle::draw() const { /* your code here */ }
```

1. Specify the **`Shape`** class as the **`public`** base class in the class header.
2. Provide an implementation for **every** pure virtual function (abstract method) in the **`Shape`** class.

For **`Triangle`** that means you **must** define a **`draw()`** member function where indicated by the comments. In reality, you'll probably do a lot more; **`Circle`** might have a **`radius`** member, the **`Square`** class could have members for the size of each **`side`**, and the **`Triangle`** class could have members for **`base`** and **`height`**.