There are 4 questions, worth a total of 40 points.

All of your files must have exactly the filename listed in the assignment. Each file should have a comment at the beginning with your name and the date. Classes should have precisely the name specifed in the assignment, should include a docstring with a description of their intended behavior, and should not contain any code that executes when the class is imported.

Blaster.py 10 points

Create a class Blaster which is a child of the Gun class. A Blaster automatically reloads itself after every shot. In other words, a Blaster is a Gun that never runs out of bullets.

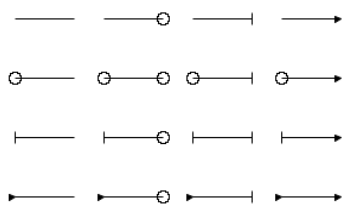
RegularPolygon.py 10 points

(This is Exercise 9.8) A *regular* polygon has all sides the same length, and all corner angles the same. Create a class RegularPolygon which is a child of the cs1graphics Polygon class. The constructor for a RegularPolygon should have the following signature:

```
__init__(self, sides, centerPoint = Point(0,0))
```

Arrow.py 10 points

Create a class Arrow which is a child of the cs1graphics Drawable class. An arrow has an initial and terminal point. The shaft of the arrow consists of a straight line from the initial to the terminal point. At the head and tail of the arrow, any of four arrowhead styles are allowed: 'none', 'circle', 'flat', and 'triangle'. These result in the arrows shown in the picture below.



You need to implement the following methods:

```
__init__(self, terminal, initial = Point(0,0),
         headstyle = 'triangle', tailstyle = 'none')
setHeadstyle(self,style)
setTailstyle(self,style)
```

Sprite.py 10 points

Create a Sprite class for cs1graphics that animates an object as it moves. The constructor for the sprite class is passed a list of Drawable objects, and optionally a speed. The class should implement three methods: setSpeed to set the speed, and the two animation functions animatedMove and animatedMoveTo which are similar to the cs1graphics move and moveTo.

For the animated motion, the Sprite should move from its current position to its destination in short steps, with a delay between each step which is controlled by the speed

parameter. At each step, the image displayed by the Sprite changes, using each image on the original list and then repeating them from the beginning.

As a hint, I made Sprite a child of the cs1graphics Layer class, and at each step removed the old frame from the Layer, then added the new frame.

You should implement the following three methods:

```
__init__(self, frames, speed=1)
     Initialize with a list of Drawable objects.  The sprite
     cycles through these when moving.  Speed is the delay in
     seconds between each step of movement.
animatedMove(self, dx, dy)
animatedMoveTo(self, x, y)
setSpeed(self,speed)
```

Here is an example of how you might use a Sprite:

```
c = Canvas()
s2 = Sprite([Square(15),Square(10),Square(5),Square(10)])
c.add(s2)
s2.animatedMove(100,100)
```