Assignment 2

Out: July 17, 1999

Due: July 31, 1999

Extension: August 4, 1999

Write a class hierarchy of 2-D geometric shapes. The entire hierarchy should support the following operations:

- public void translate(int dx, int dy)

  Translates the shape by dx horizontally and dy vertically.

- public float area()

  Computes the area of the shape.

- public boolean covers(int x, int y)

  True iff the point (x,y) is inside or on the boundary of the shape.

- public int width()

- public int height()

- public int xmin()

- public int ymin()

  These four operations together give the bounding rectangle of the shape. The bounding rectangle is the smallest rectangle covering the shape.

  If you are concerned about off-by-1 errors and such, you can play safe by giving a slightly larger rectangle. Just make sure it is reasonably small and the shape does not go outside of it.

- public void setColour(byte c)

- public byte getColour()

  A shape object should be able to remember whatever colour (a byte) the user sets to it, and return it on request.

The idea is that the following method and code will be present in the base class, `Shape`, and it should work for the entire class hierarchy because it uses only the above methods, which are supposedly properly overriden by subclass implementations.

```
public void draw(byte[][] frame) {
  int w = width(), h = height();
  int xm = xmin(), ym = ymin();
  byte c = getColour();
  for (int i = 0; i < w; i++) {
    int x = i + xm;
    if (x < 0 || x >= frame.length) continue;
    for (int j = 0; j < h; j++) {
      int y = j + ym;
      if (y < 0 || y >= frame[x].length) continue;
      if (covers(x, y)) frame[x][y] = c;
    }
  }
}
```

This algorithm renders the shape into the framebuffer (the 2-D byte array given by the parameter) by scanning every pixel inside the bounding rectangle (that is why it is ok to make it slightly larger) and only turning on those really inside the shape. It is very slow, but it should always work.

The base class should be called Shape; other classes should include Rectangle, Square, Circle, Ellipse, Triangle, and Point. They must support the following constructors, respectively:

- public Rectangle(int xmin, int ymin, int width, int height)
- public Square(int xmin, int ymin, int length)
- public Circle(int centreX, int centreY, int radius)
- public Ellipse(int centreX, int centreY, int major, int minor)
- public Point(int x, int y)
- public Triangle(int x1, int y1, int x2, int y2, int x3, int y3)

The default colour of each shape should be the number 0.

There will be a graphical user interface to stress-test your implementation.