Step 3: The Block class

Add a new class to represent and draw individual square blocks in the Tetrominos game.

WE'LL COVER THE FOLLOWING

- Task: write the constructor
- Task: write the `draw` method

Each falling piece in Tetrominos is made up of four small blocks arranged in a particular way. Let's write and test the `Block` class. Create a new file in your project called `Block.java`. To get you started, here's a skeleton of the code, which you may copy in:

Block.java | BlockSolution.java

```
import java.awt.Color;
import java.awt.Graphics;

class Block {

        public int colorIndex;

        public static Color[] colors = {Color.red, Color.blue, Color.magenta,
                        Color.orange, Color.green, Color.cyan, Color.yellow};

        public Block(int colorIndex) {
        }

        public void draw(Graphics g, int scale, int x, int y) {
        }

}
```

Task: write the constructor

Blocks come in several colors. To simplify referring to the colors, we use an array of colors, defined in the array `colors`. This array is `static` because, although we will create many blocks, there will be only one array of colors, accessible through the `Block` class.

Each block has exactly one instance variable, as we can see from the line `public int colorIndex`. If the block has a colorIndex of 2, then this would indicate that the block should be drawn using the magenta color. When a block is created using its constructor, the index of the color of the block should be passed in, and stored in the instance variable.

Write the body of the constructor, which should be only one line long. When you are done, you may check your work by looking at the second tab of the code above, `BlockSolution.java`.

Task: write the `draw` method

The Block class maintains information about the color of the block, but nothing else. Still, the Block class is a useful place to put a method that draws a block of the proper color. Write the `draw()` method, which takes as parameters an x, y coordinate in **block coordinates**, and a scaling parameter `scale`. To find the pixel coordinates, multiply `x` and `y` by `scale`, and then use the Graphics object `g` to draw the rectangle.