```
package Assign 3;
2
3
    import Media.*;
                                        // for Turtle and TurtleDisplayer
                                       // for Color objects and methods
// for math constants and functions
// for Color constants
4
    import java.awt.*;
   import static java.lang.Math.*;
import static java.awt.Color.*;
                                        // for FAST turtle
   import static Media.Turtle.*;
8
    /** This class randomly generates a cityscape with 3 to 6 buildings each containing
    5 to 15 stories.
10
      * @author Sawyer Fenwick(st# 6005011)
11
12
13
      * @version 1.0 November 2 2016
14
15 public class City {
16
      // instance variables
17
        private TurtleDisplayer display;
18
        private Turtle yertle;
19
20
        /** This constructor creates the turtle object "yertle", creates a canvas of
21
    500 \times 500 and places yertle on the
22
          * display. It calls on the "drawCityScape" method which draws the CityScape
    by calling on several other
23
          * methods. */
24
25
        public City ( ) {
26
27
        // statements including call of method
28
          yertle = new Turtle(0);
          display = new TurtleDisplayer(yertle,500,500);
29
          int buildings = (int)(3*random())+3;
30
31
          drawCityScape(buildings);
32
          display.close();
33
        }; // constructor
34
35
        /** This method creates a square
36
37
          private void drawSquare ( ) {
38
39
            // statements
40
            yertle.penDown();
41
             for (int i = 1; i <=4; i++) {
42
43
               yertle.forward(10);
               yertle.left(PI/2);
44
45
46
47
            yertle.penUp();
48
          }; // drawSquare
49
50
        /** This method draws a window built of 4 squares by calling on the
51
    "drawSquare" method.
52
          private void drawWindow ( ) {
53
54
             //statements
55
            yertle.penDown();
56
57
             for (int i = 1; i <=4; i++) {
58
               drawSquare();
59
               yertle.right(PI/2);
60
61
62
            yertle.penUp();
63
          }; // drawWindow
64
```

```
65
        /** This method draws a rectangle, which will be the outside of each building.
66
   It is passed the parameter
           "storie" which is a randomly generated number between 5 and 15. The storie
67
   is multiplied by
            30 (the height of 1 storie) to find the proper height of a rectangle that
68
   will accomodate how many
69
          * stories there will be. */
70
          private void drawRectangle(int storie) {
71
72
            int width = 70;
            int height;
73
74
            height = storie*30;
75
            yertle.penDown();
76
77
            for (int i = 1; i <= 2; i ++) {
78
              yertle.forward(width);
79
              yertle.left(PI/2);
80
              yertle.forward(height);
              yertle.left(PI/2);
81
82
83
84
           yertle.penUp();
8.5
86
          }; //drawRectangle
87
88
          /* This method draws however many stories is required to fill a building. */
89
         private void drawStorie(int storie){
90
91
          //local variables
92
            int height;
93
            height = storie*30;
94
          //statements
95
           yertle.left(PI/2);
96
            yertle.forward(15);
97
            yertle.right(PI/2);
98
            yertle.forward(20);
99
            drawWindow();
100
            yertle.forward(30);
101
            drawWindow();
102
            yertle.left(PI);
103
            yertle.forward(50);
104
            yertle.right(PI/2);
105
            for (int i = 2; i \le storie; i ++) {
106
107
              yertle.forward(30);
108
109
              yertle.right(PI/2);
              yertle.forward(20);
110
111
              drawWindow();
112
              yertle.forward(30);
113
              drawWindow();
114
              yertle.left(PI);
              yertle.forward(50);
115
116
              yertle.right(PI/2);
117
118
119
            yertle.right(PI);
120
121
            yertle.forward(height - 15);
122
            yertle.left(PI/2);
123
124
          }; // drawStorie
125
          /* This method draws a completed building, using the "drawRectangle" and
   "drawStorie" methods. */
         private void drawBuilding(int storie) {
127
128
129
            drawRectangle(storie);
1.30
            drawStorie(storie);
            C:\Users\sawye\Documents\_BrockU\COSC1P02\Assignments\Assign_3\City.java
```

```
131
132
          } // drawBuilding
133
134
          /* This method draws a complete CityScape, based on the randomly generated
   numbers for how many buildings

* will exist and how many stories each building will have. It calls on the
    "drawBuilding" method.*/
private void drawCityScape(int buildings) {
136
137
            \verb| yertle.moveTo(-210,-225); //"centering"| \\
138
139
            for(int i = 1; i <= buildings; i ++) {
140
141
              int storie = (int)(10*random())+5;
               drawBuilding(storie);
142
143
              yertle.forward(70);
144
145
146
          } // drawCityScape
147
148
          public static void main ( String[] args ) { City s = new City(); };
149
150 } // City
```