```java
1  package Ch2Task.E2_6;
2
3  /**
4   * Created by Sheldon on 10/23/2016.
5   * E2.6
6   * Replacing string object elements
7   */
8
9  public class HollePrinter {
10     public static void main(String[] args) {
11         // Construct string object
12         String sentence = "Hello, World!";
13
14         // 1st. replace string element e to $,    "H$llo, World!"
15         // 2nd. replace string element o to e,    "H$lle, Werld!"
16         // 3rd. replace string element $ to o,    "Holle, Werld!"
17         String sentenceReplace = sentence.replace("e", "$").replace("o", "e").replace("$", "o");
18
19         // Display the result
20         System.out.printf("String %25.15s%nString changed %17.15s%nExpected %23.15s%n",
21             ": " + sentence, ": " + sentenceReplace, ": " + "Holle, Werld!");
22     }
23 }
```

```java
1  package Ch2Task.E2_7;
2
3  /**
4   * Created by Sheldon on 10/23/2016.
5   * E2.7
6   * Reversing string object
7   */
8  public class ReverseTester {
9      public static void main(String[] args) {
10         // Construct string object
11         StringBuilder stringToReverse = new StringBuilder("desserts");
12         System.out.println("String to reverse: " + stringToReverse);
13
14         // Call reverse() method to reverse the string object
15         String stringReversed = stringToReverse.reverse().toString();
16
17         // Display result
18         System.out.println("String reversed: " + stringReversed);
19         System.out.println("Expected: stressed");
20     }
21 }
22
```

```java
1  package Ch2Task.E2_8;
2
3  import javax.swing.JFrame;
4  import java.awt.Color;
5
6  /**
7   * Created by Sheldon on 10/23/2016.
8   * E2.8
9   * Compare two or more colors are brighter or darker
10  */
11
12 public class BrighterDemo {
13     public static void main(String[] args) {
14         // Construct first color object
15         Color theFirstColor = new Color(50, 100, 150);
16
17         // Construct first frame
18         JFrame firstFrame = new JFrame();
19         firstFrame.setSize(200, 200);
20         firstFrame.setTitle("First Color");
21         firstFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
22         firstFrame.setVisible(true);
23
24         // Apply first color to first frame's background color
25         firstFrame.getContentPane().setBackground(theFirstColor);
26
27         // Construct second color object
28         Color theSecondColor = theFirstColor.brighter();
29
30         // Construct second frame
31         JFrame secondFrame = new JFrame();
```

```
32        secondFrame.setSize(200, 200);
33        secondFrame.setTitle("Second Color");
34        secondFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
35        secondFrame.setVisible(true);
36
37        // Apply second color to first frame's background color
38        secondFrame.getContentPane().setBackground(theSecondColor);
39
40        // Display the comparison between colors
41        System.out.println("1st Color: " + theFirstColor);
42        System.out.println("2nd Color: " + theSecondColor);
43    }
44 }
```

```
1  package Ch2Task.P2_1;
2
3  import javax.swing.JFrame;
4  import java.awt.Rectangle;
5  /**
6   * Created by Sheldon on 10/24/2016.
7   * P2.1
8   */
9  public class FourRectanglePrinter {
10     public static void main(String[] args) {
11         Rectangle box = new Rectangle(5, 10, 10, 20);
12         System.out.println(box);
13         box.translate(10, 0);
14         System.out.println(box);
15         box.translate(0, 20);
16         System.out.println(box);
17     }
18 }
19
```

```java
1  package Ch2Task.P2_3;
2
3  import javax.swing.JFrame;
4  import java.awt.Color;
5
6  /**
7   * Created by Sheldon on 10/23/2016.
8   * P2.3
9   * Displaying the intersection rectangle
10  */
11 public class IntersectionPrinter {
12     public static void main(String[] args) {
13         // Contstruct frame object
14         JFrame frame = new JFrame();
15         frame.setSize(800, 600);
16         frame.setTitle("Intersected Rectangles");
17         frame.getContentPane().setBackground(Color.WHITE);
18         frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
19         frame.setVisible(true);
20
21         // Construct rectangles
22         IntersectionRectComponent component = new IntersectionRectComponent();
23
24         // Add the components to frame
25         frame.add(component);
26     }
27 }
28
```

```java
1  package Ch2Task.P2_3;
2
3  import java.awt.Graphics;
4  import java.awt.Graphics2D;
5  import java.awt.Rectangle;
6  import java.awt.Color;
7  import javax.swing.JComponent;
8
9  /**
10  * Created by Sheldon on 10/24/2016.
11  * P2.3
12  * Construct intersection rectangle components
13  */
14 public class IntersectionRectComponent extends JComponent {
15     public void paintComponent(Graphics g) {
16         Graphics2D painter = (Graphics2D) g;      // Construct painter object
17
18         painter.setColor(Color.BLACK);
19         Rectangle r1 = new Rectangle(20, 50, 200, 300); // Construct first rectangle
20         painter.draw(r1);
21
22         painter.setColor(Color.BLUE);
23         Rectangle r2 = new Rectangle(115, 180, 300, 200); // Construct second rectangle
24 //        Rectangle r2 = new Rectangle(220, 350, 300, 200);
25 //        Rectangle r2 = new Rectangle(110, 350, 300, 200);
26 //        Rectangle r2 = new Rectangle(220, 175, 300, 200);
27         painter.draw(r2);
28
29         painter.setColor(Color.RED.darker());
30         Rectangle r3 = r1.intersection(r2); // Construct intersection rectangle
31         painter.fill(r3);
```

```java
32          float xPoint = (float) r3.getX();
33          float yPoint = (float) r3.getY();
34
35          System.out.println("Rectangle1 coordinate (x, y) : (" + r1.getX() + ", " + r1.getY() + ")");
36          System.out.println("Rectangle2 coordinate (x, y) : (" + r2.getX() + ", " + r2.getY() + ")");
37
38          // if width and height is 0, no intersection happened
39          if (r3.getWidth() == 0 && r3.getHeight() == 0) {
40              System.out.println("Intersected Rect. coordinate (x, y):(" + r3.getX() + ", " + r3.getY() + ")");
41              System.out.println("No intersection");
42          }
43          // if width is measured but height didn't, no intersection happened
44          else if (r3.getWidth() > 0 && r3.getHeight() == 0) {
45              System.out.println("Intersected Rect. coordinate (x, y):(" + r3.getX() + ", " + r3.getY() + ")");
46              System.out.println("No intersection");
47          }
48          // if height is measured but width didn't, no intersection happened
49          else if (r3.getWidth() == 0 && r3.getHeight() > 0) {
50              System.out.println("Intersected Rect. coordinate (x, y):(" + r3.getX() + ", " + r3.getY() + ")");
51              System.out.println("No intersection");
52          }
53          // if width and height are measured, intersection happened
54          else if (r3.getWidth() > 0 && r3.getHeight() > 0) {
55              System.out.println("Intersected Rect. coordinate (x, y):(" + r3.getX() + ", " + r3.getY() + ")");
56              System.out.println("Intersected Rect (w, h): (" + r3.getWidth() + ", " + r3.getHeight() + ")");
57              painter.setColor(Color.LIGHT_GRAY);
58              painter.drawString("Intersection", (xPoint + 10), (yPoint + 50));
59          }
60      }
61 }
```

```java
1  package Ch2Task.P2_5;
2
3  import java.util.Random;
4
5  /**
6   * Created by Sheldon on 10/24/2016.
7   * P2.5
8   * Generate lottery numbers array
9   */
10 public class LotteryPrinter {
11     public static void main(String[] args) {
12         int max = 50;
13         int offset = 1;
14         int range = max - offset;
15         int numbersToGenerate = 6;
16
17         Random generator = new Random();
18         System.out.println("Lottery combination: ");
19         for (int i = 0; i < numbersToGenerate; i++) {
20             int inRangeGenerator = generator.nextInt(range) + offset;
21             System.out.print(inRangeGenerator);
22             if (i < (numbersToGenerate - 1)) {
23                 System.out.print(" ");
24             }
25             if (i == (numbersToGenerate - 1)) {
26                 System.out.println("\nPlay this combination-it'll make you rich!");
27             }
28         }
29     }
30 }
31
```

```java
1  package Ch2Task.P2_8;
2
3  import javax.swing.JFrame;
4  import java.awt.Color;
5
6  /**
7   * Created by Sheldon on 10/24/2016.
8   * P2.8
9   * Measure distance from line to specified point
10  * Display the distances
11  */
12 public class LineDistanceTester {
13     public static void main(String[] args) {
14
15         JFrame myFrame = new JFrame();
16         myFrame.setSize(400, 400);
17         myFrame.setTitle("Line to Point Distance");
18         myFrame.getContentPane().setBackground(Color.WHITE);
19         myFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
20         myFrame.setVisible(true);
21
22         LineDistanceComponent calculateDistance = new LineDistanceComponent();
23         myFrame.add(calculateDistance);
24
25     }
26 }
```

```java
1  package Ch2Task.P2_8;
2
3  import javax.swing.JComponent;
4  import java.awt.Graphics2D;
5  import java.awt.Graphics;
6  import java.awt.Color;
7  import java.awt.geom.Line2D;
8  import java.text.DecimalFormat;
9
10 /**
11  * Created by Sheldon on 10/26/2016.
12  * P2.8
13  * Measure distance from line to specified point
14  * Construct line distance components
15  */
16 public class LineDistanceComponent extends JComponent {
17     public void paintComponent(Graphics g) {
18         Graphics2D painter = (Graphics2D) g;      // Construct painter object
19         Line2D.Double lineSegment = new Line2D.Double(100, 100, 200, 200);   //Construct line object
20         painter.setColor(Color.BLUE);
21         painter.draw(lineSegment);
22
23         int[] x = {100, 150, 250};   // Create x ordinates list
24         int[] y = {200, 150, 50};    // Create y ordinates list
25         int dataLength = x.length;   // Hold data length
26         double[] toPointDistance = new double[dataLength];   // Create line-to-point-distance empty list
27         double[] expectation = {70.71, 0, 141.42};
28
29         for (int i = 0; i < dataLength; i++) {
30             // Fill oval object in coordinate (x[i], y[i])
31             painter.setColor(Color.RED);
```

```
32        painter.fillOval(x[i], y[i], 5, 5);
33
34        // Calculate distance from line segment to the point specified
35        toPointDistance[i] = lineSegment.ptSegDist(x[i], y[i]);
36        String pointDistanceFormatted = new DecimalFormat(".##").format(toPointDistance[i]);
37
38        // Draw string indicates the distance
39        painter.setColor(Color.BLACK);
40        painter.drawString("Distance:" + pointDistanceFormatted, x[i], y[i]);
41
42        // Display results
43        System.out.printf("Distance between point" + (i + 1) + " and line segment: %6.2f; " +
44              "Expected: %6.2f%n", toPointDistance[i], expectation[i]);
45        }
46    }
47 }
```

```
1  package Ch2Task.E2_10;
2
3  import javax.swing.JFrame;
4  import java.awt.Color;
5
6  /**
7   * Created by Sheldon on 10/23/2016.
8   * E2.10
9   * Darken two color objects once and twice
10  */
11 public class DarkerDemo {
12     public static void main(String[] args) {
13         // Construct first color object, darken it once
14         Color theFirstColor = Color.RED.darker();
15
16         // Construct first frame
17         JFrame firstFrame = new JFrame();
18         firstFrame.setSize(200, 200);
19         firstFrame.setTitle("First Color");
20         firstFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
21         firstFrame.setVisible(true);
22
23         // Apply first color to frame's background
24         firstFrame.getContentPane().setBackground(theFirstColor);
25
26         // Construct first color object, darken it twice
27         Color theSecondColor = theFirstColor.darker();
28
29         // Construct second frame
30         JFrame secondFrame = new JFrame();
31         secondFrame.setSize(200, 200);
```

```
32      secondFrame.setTitle("Second Color");
33      secondFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
34      secondFrame.setVisible(true);
35
36      // Apply second color to frame's background
37      secondFrame.getContentPane().setBackground(theSecondColor);
38
39      // Display the result
40      System.out.println("Color darken once: " + theFirstColor);
41      System.out.println("Color darken twice: " + theSecondColor);
42  }
43 }
44
```

```java
1  package Ch2Task.E2_11;
2
3  import java.util.Random;
4
5  /**
6   * Created by Sheldon on 10/23/2016.
7   * E2.11
8   * Generate random number
9   */
10 public class DieSimulator {
11     public static void main(String[] args) {
12         // Set the upper bound and lower bound to count
13         int max = 6;
14         int offset = 1;
15         int range = max - offset;
16
17         // Construct number generator object
18         Random generator = new Random();
19
20         // Generate random number
21         int inRangeGenerator = generator.nextInt(range) + offset;
22
23         // Display result
24         System.out.println("Random number: " + inRangeGenerator);
25     }
26 }
27
```

```java
1  package Ch2Task.E2_12;
2
3  import java.text.DecimalFormat;
4  import java.util.Random;
5
6  /**
7   * Created by Sheldon on 10/23/2016.
8   * E2.12
9   * Generate random price from bound given
10  */
11 public class RandomPrice {
12     public static void main(String[] args) {
13         double maxPrice = 19.95;
14         double minPrice = 10.00;
15         double priceRange = maxPrice - minPrice;
16
17         Random generatePrice = new Random();
18         double inRangePrice = generatePrice.nextDouble() * priceRange + minPrice;
19
20         DecimalFormat rounder = new DecimalFormat(".##");
21         System.out.println("Price generated: $" + rounder.format(inRangePrice));
22     }
23 }
```

```java
1  package Ch2Task.P2_12;
2
3  import javax.swing.JFrame;
4  import java.awt.Color;
5
6  /**
7   * Created by Sheldon on 10/25/2016.
8   * P2.12
9   * Display traffic light image
10  */
11
12 public class TrafficLightViewer {
13     public static void main(String[] args) {
14         JFrame myFrame = new JFrame();
15         myFrame.setSize(700, 725);
16         myFrame.setTitle("Traffic Light");
17         myFrame.getContentPane().setBackground(Color.WHITE);
18         myFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
19         myFrame.setVisible(true);
20
21         TrafficLightComponent theTrafficLight = new TrafficLightComponent();
22         myFrame.add(theTrafficLight);
23     }
24 }
25
```

```
1  package Ch2Task.P2_12;
2
3  import java.awt.Color;
4  import java.awt.Graphics;
5  import java.awt.Graphics2D;
6  import java.awt.Rectangle;
7  import java.awt.geom.Ellipse2D;
8  import java.awt.geom.Line2D;
9  import javax.swing.JComponent;
10
11 /**
12  * Created by Sheldon on 10/25/2016.
13  * P2.12
14  * Construct traffic light components
15  */
16 public class TrafficLightComponent extends JComponent {
17     public void paintComponent(Graphics g) {
18         Graphics2D painter = (Graphics2D) g; // Construct painter object
19
20         // Draw traffic light's main body and cover
21         painter.setColor(Color.BLACK);
22         painter.fillRoundRect(350, 50, 150, 400, 20, 20);
23         painter.fillRoundRect(325, 38, 200, 15, 5, 5 );
24
25         // Draw white rectangular inside main body
26         painter.setColor(Color.WHITE);
27         painter.drawRoundRect(353, 53, 143, 393, 20, 20);
28
29         // Draw traffic light's stand
30         Rectangle trafficLightStand = new Rectangle(405, 450, 35, 200);
31         painter.setColor(Color.gray);
```

```
32      painter.fill(trafficLightStand);
33
34      // Draw the lamps
35      // RED lamp
36      Ellipse2D.Double redLamp = new Ellipse2D.Double(375, 75, 100, 100);
37      Color brightRed = Color.RED.brighter();
38      Color darkRed = Color.RED.darker();
39      painter.setColor(brightRed);
40      painter.fill(redLamp);
41      painter.setColor(Color.GRAY);
42      painter.draw(redLamp);
43      // Yellow lamp
44      Ellipse2D.Double yellowLamp = new Ellipse2D.Double(375, 200, 100, 100);
45      Color brightYellow = Color.YELLOW.brighter();
46      Color darkYellow = Color.YELLOW.darker();
47      painter.setColor(brightYellow);
48      painter.fill(yellowLamp);
49      painter.setColor(Color.GRAY);
50      painter.draw(yellowLamp);
51      // Green lamp
52      Ellipse2D.Double greenLamp = new Ellipse2D.Double(375, 325, 100, 100);
53      Color brightGreen = Color.GREEN.brighter();
54      Color darkGreen = Color.GREEN.darker();
55      painter.setColor(brightGreen);
56      painter.fill(greenLamp);
57      painter.setColor(Color.GRAY);
58      painter.draw(greenLamp);
59      // Draw lamps' cover
60      int arcWidth = 100;
61      int arcHeight = 100;
62      int arcStartAngle = 0;
```

```
63      int arcAngle = 180;
64      painter.setColor(Color.DARK_GRAY);
65      for (int i = 1; i < 30; i++) {
66          painter.drawArc(375, i + 73, arcWidth, arcHeight, arcStartAngle, arcAngle);
67          painter.drawArc(375, i + 198, arcWidth, arcHeight, arcStartAngle, arcAngle);
68          painter.drawArc(375, i + 323, arcWidth, arcHeight, arcStartAngle, arcAngle);
69      }
70  }
71 }
```