```java
package Ch2Task.E2_6;

/**
 * Created by Sheldon on 10/23/2016.
 * E2.6
 * Replacing string object elements
 */

public class HollePrinter {
    public static void main(String[] args) {
    // Construct string object
        String sentence = "Hello, World!";

    // 1st. replace string element e to $, "H$llo, World!"
    // 2nd. replace string element o to e, "H$lle, Werld!"
    // 3rd. replace string element $ to o, "Holle, Werld!"
        String sentenceReplace = sentence.replace("e", "$").replace("o", "e").replace("$", "o");

    // Display the result
        System.out.printf("String %25.15s%nString changed %17.15s%nExpected %23.15s%n",
                ": " + sentence, ": " + sentenceReplace, ": " + "Holle, Werld!");
    }
}
```

```java
package Ch2Task.E2_7;

/**
 * Created by Sheldon on 10/23/2016.
 * E2.7
 * Reversing string object
 */
public class ReverseTester {
    public static void main(String[] args) {
    // Construct string object
        StringBuilder stringToReverse = new StringBuilder("desserts");
        System.out.println("String to reverse: " + stringToReverse);

    // Call reverse() method to reverse the string object
        String stringReversed = stringToReverse.reverse().toString();

    // Display result
        System.out.println("String reversed: " + stringReversed);
        System.out.println("Expected: stressed");
    }
}

```

```java
package Ch2Task.E2_8;

import javax.swing.JFrame;
import java.awt.Color;

/**
 * Created by Sheldon on 10/23/2016.
 * E2.8
 * Compare two or more colors are brighter or darker
 */

public class BrighterDemo {
    public static void main(String[] args) {
    // Construct first color object
        Color theFirstColor = new Color(50, 100, 150);

    // Construct first frame
        JFrame firstFrame = new JFrame();
        firstFrame.setSize(200, 200);
        firstFrame.setTitle("First Color");
        firstFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        firstFrame.setVisible(true);

    // Apply first color to first frame's background color
        firstFrame.getContentPane().setBackground(theFirstColor);

    // Construct second color object
        Color theSecondColor = theFirstColor.brighter();

    // Construct second frame
        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  }
```

```java
package Ch2Task.P2_1;

import java.awt.Rectangle;
/**
 * Created by Sheldon on 10/24/2016.
 * P2.1
 */
public class FourRectanglePrinter {
    public static void main(String[] args) {
        Rectangle box = new Rectangle(5, 10, 10, 20);
        System.out.println(box);
        box.translate(10, 0);
        System.out.println(box);
        box.translate(0, 20);
        System.out.println(box);
    }
}

```

```java
package Ch2Task.P2_3;

import javax.swing.JComponent;
import javax.swing.JFrame;
import java.awt.*;

/**
 * Created by Sheldon on 10/27/2016.
 * P2.3
 * Decide wether intersection is occured or not
 * Fill the intersection area if occured
 */

public class IntersectionRectangle extends JComponent {
    public static void main(String[] args) {
    // Construct frame object
        JFrame frame = new JFrame();
        frame.setSize(600, 600);
        frame.setTitle("Intersected Rectangles");
        frame.getContentPane().setBackground(Color.WHITE);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setVisible(true);

    // Construct rectangles
        IntersectionRectangle intersectCmp = new IntersectionRectangle();

    // Add the components to frame
        frame.add(intersectCmp);
    }
    public void paintComponent(Graphics g) {
        Graphics2D painter = (Graphics2D) g;    // Construct painter object
```

```
32
33          painter.setColor(Color.BLACK);
34          Rectangle r1 = new Rectangle(20, 50, 200, 300); // Construct first rectangle
35          painter.draw(r1);
36
37          painter.setColor(Color.BLUE);
38          Rectangle r2 = new Rectangle(115, 180, 300, 200); // Construct second rectangle
39  //          Rectangle r2 = new Rectangle(220, 350, 300, 200);
40  //          Rectangle r2 = new Rectangle(110, 350, 300, 200);
41  //          Rectangle r2 = new Rectangle(220, 175, 300, 200);
42          painter.draw(r2);
43
44          painter.setColor(Color.RED.darker());
45          Rectangle r3 = r1.intersection(r2); // Construct intersection rectangle
46          painter.fill(r3);
47          float xPoint = (float) r3.getX();
48          float yPoint = (float) r3.getY();
49
50          System.out.println("Rectangle1 coordinate (x, y): (" + r1.getX() + ", " + r1.getY() + ")");
51          System.out.println("Rectangle2 coordinate (x, y): (" + r2.getX() + ", " + r2.getY() + ")");
52
53      // if width and height is 0, no intersection happened
54          if (r3.getWidth() == 0 && r3.getHeight() == 0) {
55              System.out.println("Intersected Rect. coordinate (x, y):(" + r3.getX() + ", " + r3.getY() + ")");
56              System.out.println("No intersection");
57          }
58      // if width is measured but height didn't, no intersection happened
59          else if (r3.getWidth() > 0 && r3.getHeight() == 0) {
60              System.out.println("Intersected Rect. coordinate (x, y):(" + r3.getX() + ", " + r3.getY() + ")");
61              System.out.println("No intersection");
62          }
```

```
63      // if height is measured but width didn't, no intersection happened
64          else if (r3.getWidth() == 0 && r3.getHeight() > 0) {
65              System.out.println("Intersected Rect. coordinate (x, y):(" + r3.getX() + ", " + r3.getY() + ")");
66              System.out.println("No intersection");
67          }
68      // if width and height are measured, intersection happened
69          else if (r3.getWidth() > 0 && r3.getHeight() > 0) {
70              System.out.println("Intersected Rect. coordinate (x, y):(" + r3.getX() + ", " + r3.getY() + ")");
71              System.out.println("Intersected Rect (w, h): (" + r3.getWidth() + ", " + r3.getHeight() + ")");
72              painter.setColor(Color.LIGHT_GRAY);
73              painter.drawString("Intersection", (xPoint + 10), (yPoint + 50));
74          }
75      }
76  }
77
78
```

```java
package Ch2Task.P2_5;

import java.util.Random;

/**
 * Created by Sheldon on 10/24/2016.
 * P2.5
 * Generate lottery numbers array
 */
public class LotteryPrinter {
    public static void main(String[] args) {
        int max = 50;
        int offset = 1;
        int range = max - offset;
        int numbersToGenerate = 6;

        Random generator = new Random();
        System.out.println("Lottery combination: ");
        for (int i = 0; i < numbersToGenerate; i++) {
            int inRangeGenerator = generator.nextInt(range) + offset;
            System.out.print(inRangeGenerator);
            if (i < (numbersToGenerate - 1)) {
                System.out.print(" ");
            }
            if (i == (numbersToGenerate - 1)) {
                System.out.println("\nPlay this combination-it'll make you rich!");
            }
        }
    }
}

```

```java
package Ch2Task.P2_9;

import javax.swing.*;
import java.awt.*;
import java.awt.geom.Line2D;
import java.text.DecimalFormat;

/**
 * Created by Sheldon on 10/27/2016.
 * P2.9
 * Measure and draw distance from line segment to specified point
 */
public class LineToPointDistance extends JComponent {
    public static void main(String[] args) {
        JFrame myFrame = new JFrame();
        myFrame.setSize(400, 400);
        myFrame.setTitle("Line to Point Distance");
        myFrame.getContentPane().setBackground(Color.WHITE);
        myFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        myFrame.setVisible(true);

        LineToPointDistance calculateDistance = new LineToPointDistance();
        myFrame.add(calculateDistance);
    }
    public void paintComponent(Graphics g) {
        Graphics2D painter = (Graphics2D) g;    // Construct painter object
        Line2D.Double lineSegment = new Line2D.Double(100, 100, 200, 200);  //Construct line object
        painter.setColor(Color.BLUE);
        painter.draw(lineSegment);

        int[] x = {100, 150, 250};  // Create x ordinates list
```

```
32          int[] y = {200, 150, 50};    // Create y ordinates list
33          int dataLength = x.length;  // Hold data length
34          double[] toPointDistance = new double[dataLength];  // Create line-to-point-distance empty list
35          double[] expectation = {70.71, 0, 141.42};
36
37          for (int i = 0; i < dataLength; i++) {
38              // Fill oval object in coordinate (x[i], y[i])
39              painter.setColor(Color.RED);
40              painter.fillOval(x[i], y[i], 5, 5);
41
42              // Calculate distance from line segment to the point specified
43              toPointDistance[i] = lineSegment.ptSegDist(x[i], y[i]);
44              String pointDistanceFormatted = new DecimalFormat(".##").format(toPointDistance[i]);
45
46              // Draw string indicates the distance
47              painter.setColor(Color.BLACK);
48              painter.drawString("Distance:" + pointDistanceFormatted, x[i], y[i]);
49
50              // Display results
51              System.out.printf("Distance between point" + (i + 1) + " and line segment: %6.2f; " +
52                      "Expected: %6.2f%n", toPointDistance[i], expectation[i]);
53          }
54      }
55  }
56
```

```java
package Ch2Task.E2_10;

import javax.swing.JFrame;
import java.awt.Color;

/**
 * Created by Sheldon on 10/23/2016.
 * E2.10
 * Darken two color objects once and twice
 */
public class DarkerDemo {
    public static void main(String[] args) {
    // Construct first color object, darken it once
        Color theFirstColor = Color.RED.darker();

    // Construct first frame
        JFrame firstFrame = new JFrame();
        firstFrame.setSize(200, 200);
        firstFrame.setTitle("First Color");
        firstFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        firstFrame.setVisible(true);

    // Apply first color to frame's background
        firstFrame.getContentPane().setBackground(theFirstColor);

    // Construct first color object, darken it twice
        Color theSecondColor = theFirstColor.darker();

    // Construct second frame
        JFrame secondFrame = new JFrame();
        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
package Ch2Task.E2_11;

import java.util.Random;

/**
 * Created by Sheldon on 10/23/2016.
 * E2.11
 * Generate random number
 */
public class DieSimulator {
    public static void main(String[] args) {
    // Set the upper bound and lower bound to count
        int max = 6;
        int offset = 1;
        int range = max - offset;

    // Construct number generator object
        Random generator = new Random();

    // Generate random number
        int inRangeGenerator = generator.nextInt(range) + offset;

    // Display result
        System.out.println("Random number: " + inRangeGenerator);
    }
}

```

```java
package Ch2Task.E2_12;

import java.text.DecimalFormat;
import java.util.Random;

/**
 * Created by Sheldon on 10/23/2016.
 * E2.12
 * Generate random price from bound given
 */
public class RandomPrice {
    public static void main(String[] args) {
        double maxPrice = 19.95;
        double minPrice = 10.00;
        double priceRange = maxPrice - minPrice;

        Random generatePrice = new Random();
        double inRangePrice = generatePrice.nextDouble() * priceRange + minPrice;

        DecimalFormat rounder = new DecimalFormat(".##");
        System.out.println("Price generated: $" + rounder.format(inRangePrice));
    }
}
```

```java
package Ch2Task.P2_12;

import javax.swing.*;
import java.awt.*;
import java.awt.geom.Ellipse2D;

/**
 * Created by Sheldon on 10/27/2016.
 * P2.12
 * Draws traffic light by 1 class
 */
public class TrafficLight extends JComponent {
    public static void main(String[] args) {
        JFrame myFrame = new JFrame();
        myFrame.setSize(700, 725);
        myFrame.setTitle("Traffic Light");
        myFrame.getContentPane().setBackground(Color.WHITE);
        myFrame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        myFrame.setVisible(true);

        TrafficLight theTrafficLight = new TrafficLight();
        myFrame.add(theTrafficLight);
    }
    public void paintComponent(Graphics g) {
        Graphics2D painter = (Graphics2D) g; // Construct painter object

    // Draw traffic light's main body and cover
        painter.setColor(Color.BLACK);
        painter.fillRoundRect(350, 50, 150, 400, 20, 20);
        painter.fillRoundRect(325, 38, 200, 15, 5, 5 );

```

```
32      // Draw white rectangular inside main body
33          painter.setColor(Color.WHITE);
34          painter.drawRoundRect(353, 53, 143, 393, 20, 20);
35
36      // Draw traffic light's stand
37          Rectangle trafficLightStand = new Rectangle(405, 450, 35, 200);
38          painter.setColor(Color.gray);
39          painter.fill(trafficLightStand);
40
41      // Draw the lamps
42      // RED lamp
43          Ellipse2D.Double redLamp = new Ellipse2D.Double(375, 75, 100, 100);
44          Color brightRed = Color.RED.brighter();
45          Color darkRed = Color.RED.darker();
46          painter.setColor(brightRed);
47          painter.fill(redLamp);
48          painter.setColor(Color.GRAY);
49          painter.draw(redLamp);
50      // Yellow lamp
51          Ellipse2D.Double yellowLamp = new Ellipse2D.Double(375, 200, 100, 100);
52          Color brightYellow = Color.YELLOW.brighter();
53          Color darkYellow = Color.YELLOW.darker();
54          painter.setColor(brightYellow);
55          painter.fill(yellowLamp);
56          painter.setColor(Color.GRAY);
57          painter.draw(yellowLamp);
58      // Green lamp
59          Ellipse2D.Double greenLamp = new Ellipse2D.Double(375, 325, 100, 100);
60          Color brightGreen = Color.GREEN.brighter();
61          Color darkGreen = Color.GREEN.darker();
62          painter.setColor(brightGreen);
```

```
63          painter.fill(greenLamp);
64          painter.setColor(Color.GRAY);
65          painter.draw(greenLamp);
66      // Draw lamps' cover
67          int arcWidth = 100;
68          int arcHeight = 100;
69          int arcStartAngle = 0;
70          int arcAngle = 180;
71          painter.setColor(Color.DARK_GRAY);
72          for (int i = 1; i < 30; i++) {
73              painter.drawArc(375, i + 73, arcWidth, arcHeight, arcStartAngle, arcAngle);
74              painter.drawArc(375, i + 198, arcWidth, arcHeight, arcStartAngle, arcAngle);
75              painter.drawArc(375, i + 323, arcWidth, arcHeight, arcStartAngle, arcAngle);
76          }
77      }
78  }
79
```