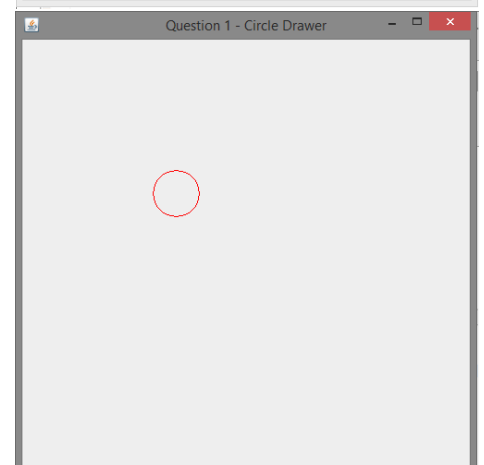
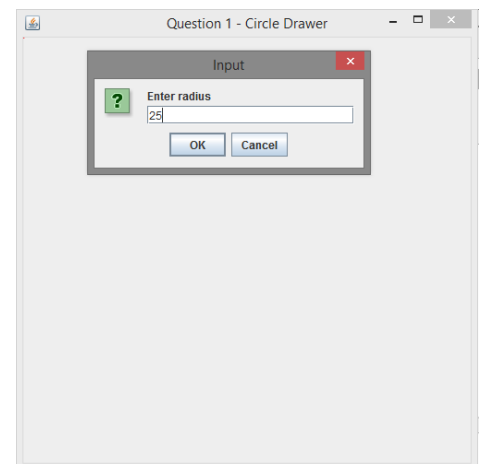


1.

```
CircleDrawerComponent.java  CircleDrawerFrame.java  X
1  /**
2   * Devante Wilson
3   * November 20, 2015
4   *
5   * Question 1
6   *
7   * Program allows user to specify
8   * a circle by clicking on the center
9   * and typing the radius
10  *
11  * Class designs a JFrame
12  */
13
14  // import classes
15  import javax.swing.JFrame;
16  import java.awt.event.MouseEvent;
17  import java.awt.event.MouseListener;
18  import java.awt.Dimension;
19  import javax.swing.JOptionPane;
20
21  public class CircleDrawerFrame extends JFrame
22  {
23      // specify instance and class variables
24      private CircleDrawerComponent component;
25      public int radius;
26      static final int FRAME_WIDTH = 500, FRAME_HEIGHT= 500;
27
28      // constructor
29      CircleDrawerFrame()
30      {
31          component = new CircleDrawerComponent();
32          component.setPreferredSize(new Dimension(400,350));
33
34          class MousePressListener implements MouseListener
35          {
36              // determine if mouse was pressed
37              public void mousePressed(MouseEvent e)
38              {
39                  // define variables/objects
40                  int x = e.getX();
41                  int y = e.getY();
42                  String ansRadius = JOptionPane.showInputDialog("Enter radius"); // prompt user
43
44                  radius = Integer.parseInt(ansRadius); // radius of ellipse
45                  component.setPositionAndSize(x,y,radius); // positioning
46              }
47
48              // unaltered methods from MouseListener
49              public void mouseClicked(MouseEvent e) {}
50
51              public void mouseReleased(MouseEvent e) {}
52
53              public void mouseEntered(MouseEvent e) {}
54
55              public void mouseExited(MouseEvent e) {}
56          }
57
58          MousePressListener listener = new MousePressListener(); // new listener object
59          component.addMouseListener(listener); // add listener to component
60          setSize(FRAME_WIDTH, FRAME_HEIGHT); // set size of frame
61          add(component); // add component to frame
62      }
63  }
```

```
CircleDrawerComponent.java  CircleDrawerFrame.java  CircleDrawerTest.java
2 * Devante Wilson
15 import java.awt.Color;
16 import java.awt.Graphics;
17 import java.awt.Graphics2D;
18 import javax.swing.JComponent;
19
20 import java.awt.geom.Ellipse2D;
21
22 public class CircleDrawerComponent extends JComponent
23 {
24     // specify instance variables
25     private int x, y, radius;
26
27     // mutator method
28     public void setPositionAndSize(int x, int y, int radius)
29     {
30         this.x = x;
31         this.y = y;
32         this.radius = radius;
33     }
34
35     // draw the circle
36     public void paintComponent(Graphics g)
37     {
38         super.paintComponent(g);
39         Graphics2D g2 = (Graphics2D) g;
40
41         Ellipse2D.Double ellipse
42             = new Ellipse2D.Double(x - radius, y - radius, radius * 2, radius * 2);
43
44         // set color
45         g2.setColor(Color.red);
46         // draw onto graphics context
47         g2.draw(ellipse);
48
49         // call repaint as visual properties were altered
50         repaint();
51     }
52
53     // default no parameter constructor
54     public CircleDrawerComponent()
55     {
56
57     }
58 }
```

```
CircleDrawerComponent.java  CircleDrawerFrame.java  CircleDrawerTest.java
2 * Devante Wilson
15 import javax.swing.JFrame;
16
17 public class CircleDrawerTest
18 {
19     public static void main(String[] args)
20     {
21         // define objects
22         JFrame frame = new CircleDrawerFrame();
23
24         // set frame properties
25         frame.setTitle("Question 1 - Circle Drawer");
26         frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
27         frame.setVisible(true);
28     }
29 }
```



2.

```
Employee.java  Manager.java  Executive.java  Q2Tester.java  R
1 /**
2  * Devante Wilson
3  * November 21, 2015
4  *
5  * Question 2
6  *
7  * Program describes a set of
8  * entities of a company such as
9  * an employee, manager and executive.
10 */
11
12 public class Employee
13 {
14     // instance variables
15     private String name;
16     private double salary;
17
18     // constructor/mutator method
19     public Employee(String name, double salary)
20     {
21         this.name = name;
22         this.salary = salary;
23     }
24
25     // retrieve info.
26     public String toString()
27     {
28         return "name: " + name +
29             "\nsalary: " + salary +
30             "\n";
31     }
32
33     // default no parameter constructor
34     public Employee()
35     {
36     }
37 }
38 }

Employee.java  Manager.java  Executive.java  Q2Tester.java  RandomShapeG...  RandomSh
11
12 public class Manager extends Employee
13 {
14     // instance variable(s)
15     private String department;
16
17     // constructor/mutator method
18     public Manager(String name, double salary, String department)
19     {
20         super(name, salary);
21         this.department = department;
22     }
23
24     // retrieve info.
25     public String toString()
26     {
27         return super.toString() + department + "\n";
28     }
29
30     // default no parameter constructor
31     public Manager()
32     {
33     }
34 }
35 }

Employee.java  Manager.java  Executive.java  Q2Tester.java  RandomShapeG...  RandomSh
1 /**
2  * Devante Wilson
3  * November 21, 2015
4  *
5  * Question 2
6  *
7  * Program describes a set of
8  * entities of a company such as
9  * an employee, manager and executive.
10 */
11
12 public class Executive extends Manager
13 {
14     // constructor/mutator method
15     public Executive(String name, double salary, String department)
16     {
17         super(name, salary, department);
18     }
19
20     // retrieve info.
21     public String toString()
22     {
23         return super.toString();
24     }
25
26     // default no parameter constructor
27     public Executive()
28     {
29     }
30 }
31 }
```

```
Employee.java | Manager.java | Executive.java | Q2Tester.java | RandomShapeG... | RandomShapeC... | »
2 * Devante Wilson
13
14 public class Q2Tester
15 {
16     public static void main(String[] args)
17     {
18         // create objects
19         Employee emp = new Employee("Employee name", 1);
20         Manager ma = new Manager("Manager name", 2, "Manager department");
21         Executive ex = new Executive("Executive name", 3, "Executive department");
22
23         // call methods and print results
24         System.out.println(emp.toString()); // employee info.
25         System.out.println(ma.toString()); // manager info.
26         System.out.println(ex.toString()); // executive info.
27     }
28 }
```

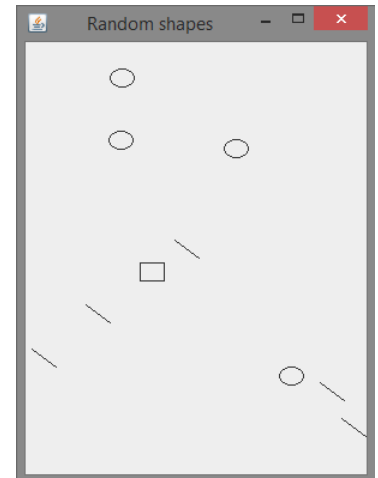
```
<terminated> Q2Tester [Java Application]
name: Employee name
salary: 1.0
|
name: Manager name
salary: 2.0
Manager department
|
name: Executive name
salary: 3.0
Executive department
```

3.

```
RandomShapeGe... | RandomShapeCo... | RandomShapeVie... | CircleDrawerC... | CircleDrawerF... | CircleD
2 * Devante Wilson
12 import java.awt.Rectangle;
17
18 public class RandomShapeGenerator
19 {
20     // declare class variables
21     static Random generator;
22     static int windowW, windowH;
23
24     // constructor
25     public RandomShapeGenerator(int w, int h)
26     {
27         // set random generator
28         generator = new Random();
29
30         // set window width and height
31         windowW = w;
32         windowH = h;
33     }
34
35     /* generate a random shape
36     * @return a shape: rectangle, ellipse, or line
37     */
38     public Shape randomShape()
39     {
40         int s = generator.nextInt(3); // hold random shape
41         int x = generator.nextInt(windowW); // horizontal boundary is window width
42         int y = generator.nextInt(windowH); // vertical boundary is window height
43
44         // dimensions of shapes
45         final int WIDTH = 20;
46         final int HEIGHT = 15;
47
48         if (s == 0) // return a rectangle
49         {
50             return new Rectangle(x, y, WIDTH, HEIGHT);
51         }
52         else if (s == 1) // return an ellipse
53         {
54             return new Ellipse2D.Double(x, y, WIDTH, HEIGHT);
55         }
56         else // return a line
57         {
58             // compensate for x2, y2, coordinates (as if they were length/width)
59             return new Line2D.Double(x, y, x + WIDTH, y + HEIGHT);
60         }
61     }
62 }
```

```
RandomShapeGe... RandomShapeCo... RandomShapeVie... CircleDrawerC... CircleDrawerF... CircleDrawe
2 * Devante Wilson
12 import java.awt.Color;
16
17 public class RandomShapeComponent extends JComponent
18 {
19     // draw the random shape
20     public void paintComponent(Graphics g)
21     {
22         // define objects
23         Graphics2D g2 = (Graphics2D) g;
24         RandomShapeGenerator rsg = new RandomShapeGenerator(getWidth(), getHeight());
25
26         // call ten times
27         for (int i = 10; i > 0; i--)
28         {
29             g2.draw(rsg.randomShape());
30         }
31     }
32 }
```

```
RandomShapeGe... RandomShapeCo... RandomShapeVie... CircleDrawerC... CircleDrawe
1 /**
2  * Devante Wilson
3  * November 21, 2015
4  *
5  * Question 3
6  *
7  * Generates random objects by
8  * implementing the Shape interface
9  */
10
11 // import class
12 import javax.swing.JFrame;
13
14 public class RandomShapeViewer
15 {
16     public static void main(String[] args)
17     {
18         // define objects
19         JFrame frame = new JFrame();
20         RandomShapeComponent component = new RandomShapeComponent();
21
22         // set frame properties
23         frame.setSize(300, 400);
24         frame.setTitle("Random shapes");
25         frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
26         frame.add(component);
27         frame.setVisible(true);
28     }
29 }
```



4.

```
1  /**
2   * Devante Wilson
3   * November 21, 2015
4   *
5   * Question 4
6   *
7   * Program reads a file containing
8   * two columns of floating-point numbers
9   * and prints average of each column
10  */
11
12  // import classes
13  import java.io.File;
14  import java.io.FileNotFoundException;
15  import java.util.Scanner;
16
17  public class AverageColumns
18  {
19      public static void main(String[] args) throws FileNotFoundException
20      {
21          // define new Scanner object
22          Scanner in = new Scanner(System.in);
23
24          // prompt user for file name
25          System.out.print("Please enter the file name: ");
26          String filename = in.next();
27
28          // close Scanner object
29          in.close();
30
31          // attempt to read from file
32          try
33          {
34              File file = new File(filename);
35
36              if (!file.exists())
37                  throw new FileNotFoundException();
38          }
39          catch (FileNotFoundException e)
40          {
41              System.err.println("Warning: " + e.getMessage());
42          }
43
44          // define variables/objects
45          Scanner scan = new Scanner(new File(filename));
46          double sum1 = 0, sum2 = 0;
47          int count = 0;
48          String[] line;
49
50          while (scan.hasNextDouble())
51          {
52              // define array to hold columns
53              line = scan.nextLine().split("\\s+"); // one or more whitespaces
54              sum1 += Double.parseDouble(line[0]);
55              sum2 += Double.parseDouble(line[1]);
56              count = line.length;
57          }
58
59          // print average of column 1
60          System.out.println("Average of column 1: " + sum1 / count);
61
62          // print average of column 2
63          System.out.println("Average of column 2: " + sum2 / count);
64
65          // close Scanner object
66          scan.close();
67      }
68  }
69
```

testfile - Notepad

File Edit Format View Help

1.5 2.5
10.5 4.5

Console

<terminated> AverageColumns [Java Application] C:\Software\JB
Please enter the file name: testfile
Average of column 1: 6.0
Average of column 2: 3.5

5.

```
RandomShapeV... AverageColum... MathGame.java MathGameRunn...
1 /**
2  * Devante Wilson
3  * November 25, 2015
4  *
5  * Question 5
6  *
7  * Program teaches arithmetic to a young child.
8  * Testing only on addition and subtraction.
9  * Generating random problems including 3 levels.
10 *
11 * Class creates the game.
12 */
13
14 // import classes
15 import java.util.Scanner;
16 import java.util.InputMismatchException;
17 import java.util.Random;
18
19 public class MathGame
20 {
21     // instance variables
22     private Player player;
23     private Scanner in;
24
25     // constructor
26     public MathGame()
27     {
28         player = new Player();
29         in = new Scanner(System.in);
30     }
31
32     // play the game
33     public void play()
34     {
35         readPlayerInformation();
36         String response = "";
37         boolean done = false;
38
39         while (!done)
40         {
41             playRound();
42             System.out.print("Do you want to play again? (Y/N) ");
43             response = in.next();
44
45             if (!response.equalsIgnoreCase("y"))
46             {
47                 done = true;
48             }
49         }
50     }
}
```

```
RandomShapeV... AverageColumn... MathGame.java MathGameRunn... Player.java
52 // read user information
53 public void readPlayerInformation()
54 {
55     String name = "";
56     System.out.print("What is your name? ");
57     name = in.next();
58     int initialLevel = 0;
59     boolean done = false;
60
61     while (!done)
62     {
63         System.out.print("At what level do you want to start? (1-3) ");
64
65         try
66         {
67             initialLevel = in.nextInt();
68
69             if (initialLevel >= 1 && initialLevel <= 3)
70             {
71                 done = true;
72             }
73         }
74         catch (InputMismatchException e)
75         {
76             in.next(); // read the newline character
77             System.out.println("Plese, enter a number between a 1 and 3.");
78         }
79     }
80
81     player = new Player(name, initialLevel);
82 }
83
```

```
RandomShapeV... AverageColumn... MathGame.java MathGameRunn...
84 // play one round of the game
85 public void playRound()
86 {
87     int i1 = 0;
88     int i2 = 0;
89     int answer = 0;
90     Random generator = new Random();
91
92     if (player.getLevel() == 1)
93     {
94         boolean done = false;
95
96         while (!done)
97         {
98             i1 = generator.nextInt(9) + 1;
99             i2 = generator.nextInt(9) + 1;
100             answer = i1 + i2;
101
102             if (answer < 9)
103             {
104                 done = true;
105             }
106         }
107
108         System.out.print("What is " + i1 + " + " + i2 + " ? ");
109     }
110     else if (player.getLevel() == 2)
111     {
112         i1 = generator.nextInt(9) + 1;
113         i2 = generator.nextInt(9) + 1;
114         answer = i1 + i2;
115         System.out.print("What is " + i1 + " + " + i2 + " ? ");
116     }
117
118     if (player.getLevel() == 3)
119     {
120         boolean done = false;
121
122         while (!done)
123         {
124             i1 = generator.nextInt(9) + 1;
125             i2 = generator.nextInt(9) + 1;
126             answer = i2 - i1;
127
128             if (answer > 0)
129             {
130                 done = true;
131             }
132         }
133
134         System.out.print("What is " + i2 + " - " + i1 + " ? ");
135     }
136
137     System.out.println(getResults(answer));
138 }
```



```
RandomShapeV... AverageColumn... MathGame.java MathGameRun... Player.java GraphData.java »
140  /*
141  * get results of the play.
142  * @param answer the answer
143  * @return the results
144  */
145  public String getResults(int answer)
146  {
147      int guess = getGuess();
148
149      if (answer != guess) //Allow a second try.
150      {
151          System.out.println("Sorry, that is incorrect. Please try one more time.");
152          guess = getGuess();
153      }
154
155      String result = "";
156
157      if (answer == guess)
158      {
159          result = "Congratulations, " + player.getName() + "! That is correct.";
160          player.incrementScore();
161      }
162      else
163      {
164          result = "Sorry, " + player.getName() +
165                  ". The correct answer is " + answer;
166      }
167
168      return result;
169  }
```

```
RandomShapeV... AverageColumn... MathGame.java MathGameRun... Player.java GraphData.java
170
171  /*
172  * get the guess from the player
173  * @param the guess from the player.
174  */
175  public int getGuess()
176  {
177      int guess = 0;
178      boolean done = false;
179      String input = "";
180
181      while (!done)
182      {
183          try
184          {
185              if (player.getLevel() == 3)
186              {
187                  System.out.print("Please enter the difference: ");
188                  guess = in.nextInt();
189              }
190              else
191              {
192                  System.out.print("Please enter the sum: ");
193                  guess = in.nextInt();
194              }
195
196              if (guess > 0)
197              {
198                  done = true;
199              }
200          }
201          catch (InputMismatchException e)
202          {
203              in.next(); // read the newline character
204              System.out.println("The response must be a number.");
205          }
206      }
207
208      return guess;
209  }
210 }
```

```
11 public class Player
12 {
13     // instance variables
14     private String name;
15     private int score;
16     private int level;
17
18     // constructor
19     public Player()
20     {
21         name = "";
22         score = 0;
23         level = 1;
24     }
25
26     /* construct a Player object
27      * with the player's name and initial level.
28      * @param playerName the player's name
29      * @param initialLevel the initial level
30      */
31     public Player(String playerName, int initialLevel)
32     {
33         name = playerName;
34         score = 0;
35         level = initialLevel;
36     }
37
38     // increment the player's score.
39     public void incrementScore()
40     {
41         score++;
42
43         if (score % 5 == 0 && level < 3)
44         {
45             level++;
46         }
47     }
48
49     /* get the current level.
50      * @return level the current level
51      */
52     public int getLevel()
53     {
54         return level;
55     }
56
57     /*
58      * get the name of the player.
59      * @return name the name of the player;
60      */
61     public String getName()
62     {
63         return name;
64     }
65 }
```

```
1 /**
2  * Devante Wilson
3  * November 25, 2015
4  *
5  * Question 5
6  *
7  * This class runs the math game.
8  */
9
10 public class MathGameRunner
11 {
12     public static void main(String[] args)
13     {
14         MathGame game = new MathGame();
15         game.play();
16     }
17 }
```

```
C:\Users\100554361\Documents\Object Oriented Programming and Design\Eclipse Work
space\OOP - Assignment 4 - Question 5\src>java -jar gameSeal.jar
What is your name? Devante
At what level do you want to start? <1-3> 1
What is 6 + 1 ? Please enter the sum: 7
Congratulations, Devante! That is correct.
Do you want to play again? <Y/N> y
What is 1 + 2 ? Please enter the sum: 3
Congratulations, Devante! That is correct.
Do you want to play again? <Y/N> y
What is 4 + 2 ? Please enter the sum: 0
Please enter the sum: 1
Sorry, that is incorrect. Please try one more time.
Please enter the sum: 3
Sorry, Devante. The correct answer is 6
Do you want to play again? <Y/N> n
```

Manifest.txt - Notepad

File Edit Format View Help

Main-Class: MathGameRunner