Chapter 3 Sample Programs

```
//********************************************************************
//  Age.java       Author: Lewis and Loftus
//  Demonstrates the use of an if statement.
//********************************************************************
import cs1.Keyboard;
public class Age
{
   //-----------------------------------------------------------------
   //  Reads the user's age and prints comments accordingly.
   //-----------------------------------------------------------------
   public static void main (String[] args)
   {
      final int MINOR = 21;

      System.out.print ("Enter your age: ");
      int age = Keyboard.readInt();
      System.out.println ("You entered: " + age);
      if (age < MINOR)
         System.out.println ("Youth is a wonderful thing. Enjoy.");
      System.out.println ("Age is a state of mind.");
   }
}

//********************************************************************
//  Wages.java       Author: Lewis and Loftus
//  Demonstrates the use of an if-else statement.
//********************************************************************
import java.text.NumberFormat;
import cs1.Keyboard;
public class Wages
{
   //-----------------------------------------------------------------
   //  Reads the number of hours worked and calculates wages.
   //-----------------------------------------------------------------
   public static void main (String[] args)
   {
      final double RATE = 8.25;              // regular pay rate
      final int STANDARD = 40;               // standard hours in a work week
      double pay = 0.0;

      System.out.print ("Enter the number of hours worked: ");
      int hours = Keyboard.readInt();
      System.out.println ();
      // Pay overtime at "time and a half"
      if (hours > STANDARD)
         pay = STANDARD * RATE + (hours-STANDARD) * (RATE * 1.5);
      else
         pay = hours * RATE;

      NumberFormat fmt = NumberFormat.getCurrencyInstance();
      System.out.println ("Gross earnings: " + fmt.format(pay));
   }
}
```

```
//********************************************************************
//  MinOfThree.java       Author: Lewis and Loftus
//  Demonstrates the use of nested if statements.
//********************************************************************
import cs1.Keyboard;

public class MinOfThree
{
   //-----------------------------------------------------------------
   //  Reads three integers from the user and determines the smallest value.
   //-----------------------------------------------------------------
   public static void main (String[] args)
   {
      int num1, num2, num3, min = 0;

      System.out.println ("Enter three integers: ");
      num1 = Keyboard.readInt();
      num2 = Keyboard.readInt();
      num3 = Keyboard.readInt();

      if (num1 < num2)
         if (num1 < num3)
            min = num1;
         else
            min = num3;
      else
         if (num2 < num3)
            min = num2;
         else
            min = num3;

      System.out.println ("Minimum value: " + min);
   }
}

//********************************************************************
//  GradeReport.java       Author: Lewis and Loftus
//  Demonstrates the use of a switch statement.
//********************************************************************
import cs1.Keyboard;

public class GradeReport
{
   //-----------------------------------------------------------------
   //  Reads a grade from the user and prints comments accordingly.
   //-----------------------------------------------------------------
   public static void main (String[] args)
   {
      int grade, category;

      System.out.print ("Enter a numeric grade (0 to 100): ");
      grade = Keyboard.readInt();

      category = grade / 10;

      System.out.print ("That grade is ");
```

```
      switch (category)
      {
        case 10:
          System.out.println ("a perfect score. Well done.");
          break;
        case 9:
          System.out.println ("well above average. Excellent.");
          break;
        case 8:
          System.out.println ("above average. Nice job.");
          break;
        case 7:
          System.out.println ("average.");
          break;
        case 6:
          System.out.println ("below average. You should see the");
          System.out.println ("instructor to clarify the material "
                              + "presented in class.");
          break;
        default:
          System.out.println ("not passing.");
      }
    }
}

//********************************************************************
//  Counter.java       Author: Lewis and Loftus
//  Demonstrates the use of a while loop.
//********************************************************************

public class Counter
{
  //-----------------------------------------------------------------
  //  Prints integer values from 1 to a specific limit.
  //-----------------------------------------------------------------
  public static void main (String[] args)
  {
    final int LIMIT = 5;
    int count = 1;

    while (count <= LIMIT)
    {
      System.out.println (count);
      count = count + 1;
    }

    System.out.println ("Done");
  }
}
```

```
//********************************************************************
//  Average.java       Author: Lewis and Loftus
//  Demonstrates the use of a while loop, a sentinel value, and a
//  running sum.
//********************************************************************

import java.text.DecimalFormat;
import cs1.Keyboard;

public class Average
{
   //-----------------------------------------------------------------
   //  Computes the average of a set of values entered by the user.
   //  The running sum is printed as the numbers are entered.
   //-----------------------------------------------------------------
   public static void main (String[] args)
   {
      int sum = 0, value, count = 0;
      double average;

      System.out.print ("Enter an integer (0 to quit): ");
      value = Keyboard.readInt();

      while (value != 0)                              // sentinel value of 0 to terminate loop
      {
         count++;

         sum += value;
         System.out.println ("The sum so far is " + sum);

         System.out.print ("Enter an integer (0 to quit): ");
         value = Keyboard.readInt();
      }

      System.out.println ();
      System.out.println ("Number of values entered: " + count);

      average = (double)sum / count;

      DecimalFormat fmt = new DecimalFormat ("0.###");
      System.out.println ("The average is " + fmt.format(average));
   }
}
```

```
//********************************************************************
//  WinPercentage.java       Author: Lewis and Loftus
//  Demonstrates the use of a while loop for input validation.
//********************************************************************

import java.text.NumberFormat;
import cs1.Keyboard;
public class WinPercentage
{
   //-----------------------------------------------------------------
   //  Computes the percentage of games won by a team.
   //-----------------------------------------------------------------
   public static void main (String[] args)
   {
      final int NUM_GAMES = 12;
      int won;
      double ratio;

      System.out.print ("Enter the number of games won (0 to "
                        + NUM_GAMES + "): ");
      won = Keyboard.readInt();
      while (won < 0 || won > NUM_GAMES)
      {
         System.out.print ("Invalid input. Please reenter: ");
         won = Keyboard.readInt();
      }

      ratio = (double)won / NUM_GAMES;

      NumberFormat fmt = NumberFormat.getPercentInstance();
      System.out.println ();
      System.out.println ("Winning percentage: " + fmt.format(ratio));
   }
}

//********************************************************************
//  Forever.java       Author: Lewis and Loftus
//  Demonstrates an INFINITE LOOP.  WARNING!!
//********************************************************************
public class Forever
{
   //-----------------------------------------------------------------
   //  Prints ever decreasing integers in an INFINITE LOOP!
   //-----------------------------------------------------------------
   public static void main (String[] args)
   {
      int count = 1;
      while (count <= 25)
      {
         System.out.println (count);
         count = count - 1;
      }
      System.out.println ("Done");                    // this statement is never reached
   }
}
```

```
//********************************************************************
//  PalindromeTester.java       Author: Lewis and Loftus
//  Demonstrates the use of nested while loops.
//********************************************************************

import cs1.Keyboard;

public class PalindromeTester
{
  //-----------------------------------------------------------------
  //  Tests strings to see if they are palindromes.
  //-----------------------------------------------------------------
  public static void main (String[] args)
  {
    String str, another = "y";
    int left, right;

    while (another.equalsIgnoreCase("y"))          // allows y or Y
    {
      System.out.println ("Enter a potential palindrome:");
      str = Keyboard.readString();

      left = 0;
      right = str.length() - 1;

      while (str.charAt(left) == str.charAt(right) && left < right)
      {
        left++;
        right--;
      }

      System.out.println();

      if (left < right)
        System.out.println ("That string is NOT a palindrome.");
      else
        System.out.println ("That string IS a palindrome.");

      System.out.println();
      System.out.print ("Test another palindrome (y/n)? ");
      another = Keyboard.readString();
    }
  }
}
```

```
//********************************************************************
//  Counter2.java       Author: Lewis and Loftus
//  Demonstrates the use of a do loop.
//********************************************************************
public class Counter2
{
   //-----------------------------------------------------------------
   //  Prints integer values from 1 to a specific limit.
   //-----------------------------------------------------------------
   public static void main (String[] args)
   {
      final int LIMIT = 5;
      int count = 0;
      do {
         count = count + 1;
         System.out.println (count);
      }
      while (count < LIMIT);
      System.out.println ("Done");
   }
}

//********************************************************************
//  Multiples.java       Author: Lewis and Loftus
//  Demonstrates the use of a for loop.
//********************************************************************
import cs1.Keyboard;
public class Multiples
{
   //-----------------------------------------------------------------
   //  Prints multiples of a user-specified number up to a user-
   //  specified limit.
   //-----------------------------------------------------------------
   public static void main (String[] args)
   {
      final int PER_LINE = 5;
      int value, limit, mult, count = 0;

      System.out.print ("Enter a positive value: ");
      value = Keyboard.readInt();
      System.out.print ("Enter an upper limit: ");
      limit = Keyboard.readInt();
      System.out.println ();
      System.out.println ("The multiples of " + value + " between " +
                          value + " and " + limit + " (inclusive) are:");

      for (mult = value; mult <= limit; mult += value) {
         System.out.print (mult + "\t");
         // Print a specific number of values per line of output
         count++;
         if (count % PER_LINE == 0)
            System.out.println();
      }
   }
}
```

```
//********************************************************************
//  Stars.java       Author: Lewis and Loftus
//  Demonstrates the use of nested for loops.
//********************************************************************
public class Stars
{
   //-----------------------------------------------------------------
   //  Prints a triangle shape using asterisk (star) characters.
   //-----------------------------------------------------------------
   public static void main (String[] args)
   {
      final int MAX_ROWS = 10;
      for (int row = 1; row <= MAX_ROWS; row++)
      {
         for (int star = 1; star <= row; star++)
            System.out.print ("*");
         System.out.println();
      }
   }
}
```

```
//********************************************************************
//  ExamGrades.java       Author: Lewis and Loftus
//  Demonstrates the use of various control structures.
//********************************************************************
import java.text.DecimalFormat;
import cs1.Keyboard;

public class ExamGrades
{
   //-----------------------------------------------------------------
   //  Computes the average, minimum, and maximum of a set of exam
   //  scores entered by the user.
   //-----------------------------------------------------------------
   public static void main (String[] args)
   {
      int grade, count = 0, sum = 0, max, min;
      double average;

      //  Get the first grade and give min and max that initial value
      System.out.print ("Enter the first grade (-1 to quit): ");
      grade = Keyboard.readInt();
      max = min = grade;
      //  Read and process the rest of the grades
      while (grade >= 0)
      {
         count++;
         sum += grade;

         if (grade > max)     max = grade;
         else
            if (grade < min)   min = grade;

         System.out.print ("Enter the next grade (-1 to quit): ");
         grade = Keyboard.readInt ();
      }
```

```
      if (count == 0)
        System.out.println ("No valid grades were entered.");
      else
      {
        DecimalFormat fmt = new DecimalFormat ("0.##");
        average = (double)sum / count;
        System.out.println();
        System.out.println ("Total number of students: " + count);
        System.out.println ("Average grade: " + fmt.format(average));
        System.out.println ("Highest grade: " + max);
        System.out.println ("Lowest grade: " + min);
      }
    }
}

//********************************************************************
//  Bullseye.java       Author: Lewis and Loftus
//  Demonstrates the use of conditionals and loops to guide drawing.
//********************************************************************
import java.applet.Applet;
import java.awt.*;

public class Bullseye extends Applet
{
  private final int MAX_WIDTH = 300;
  private final int NUM_RINGS = 5;
  private final int RING_WIDTH = 25;

  //-----------------------------------------------------------------
  //  Paints a bullseye target.
  //-----------------------------------------------------------------
  public void paint (Graphics page)
  {
    int x = 0, y = 0, diameter;
    setBackground (Color.cyan);
    diameter = MAX_WIDTH;
    page.setColor (Color.white);

    for (int count = 0; count < NUM_RINGS; count++)
    {
      if (page.getColor() == Color.black)                    // alternate colors
        page.setColor (Color.white);
      else
        page.setColor (Color.black);

      page.fillOval (x, y, diameter, diameter);

      diameter -= (2 * RING_WIDTH);
      x += RING_WIDTH;
      y += RING_WIDTH;
    }
    // Draw the red bullseye in the center
    page.setColor (Color.red);
    page.fillOval (x, y, diameter, diameter);
  }
}
```

```
<! Bullseye.html>
<HTML>

<HEAD>
<TITLE>The Bullseye Applet</TITLE>
</HEAD>

<BODY>
<center>

<H3>The Bullseye Applet:</H3>

<APPLET CODE="Bullseye.class" WIDTH=300 HEIGHT=300>
</APPLET>

<HR>
</center>
</BODY>
</HTML>


//********************************************************************
//  Boxes.java       Author: Lewis and Loftus
//  Demonstrates the use of conditionals and loops to guide drawing.
//********************************************************************
import java.applet.Applet;
import java.awt.*;

public class Boxes extends Applet
{
   private final int NUM_BOXES = 50;
   private final int THICKNESS = 5;
   private final int MAX_SIDE = 50;
   private final int MAX_X = 350;
   private final int MAX_Y = 250;

   //-----------------------------------------------------------------
   //  Paints boxes of random width and height in a random location.
   //  Narrow or short boxes are highlighted with a fill color.
   //-----------------------------------------------------------------
   public void paint(Graphics page)
   {
      int x, y, width, height;

      setBackground (Color.black);

      for (int count = 0; count < NUM_BOXES; count++)
      {
         x = (int) (Math.random() * MAX_X);
         y = (int) (Math.random() * MAX_Y);

         width = (int) (Math.random() * MAX_SIDE);
         height = (int) (Math.random() * MAX_SIDE);
```

```
        if (width <= THICKNESS)                 // check for narrow box
        {
          page.setColor (Color.yellow);
          page.fillRect (x, y, width, height);
        }
        else
          if (height <= THICKNESS)              // check for short box
          {
            page.setColor (Color.green);
            page.fillRect (x, y, width, height);
          }
          else
          {
            page.setColor (Color.white);
            page.drawRect (x, y, width, height);
          }
      }
    }
}

<! Boxes.html>
<HTML>
<HEAD>
<TITLE>The Boxes Applet</TITLE>
</HEAD>

<BODY>
<center>

<H3>The Boxes Applet:</H3>
<APPLET CODE="Boxes.class" WIDTH=400 HEIGHT=300></APPLET>
<HR>
</center>
</BODY>
</HTML>


//********************************************************************
//  BarHeights.java       Author: Lewis and Loftus
//  Demonstrates the use of conditionals and loops to guide drawing.
//********************************************************************
import java.applet.Applet;
import java.awt.*;

public class BarHeights extends Applet
{
  private final int NUM_BARS = 10;
  private final int BAR_WIDTH = 30;
  private final int MAX_HEIGHT = 300;
  private final int GAP = 9;
```

```
  //-----------------------------------------------------------------
  //  Paints bars of varying heights, tracking the tallest and
  //  shortest bars, which are redrawn in color at the end.
  //-----------------------------------------------------------------
  public void paint (Graphics page)
  {
    int x, height;
    int tallX = 0, tallest = 0, shortX = 0, shortest = MAX_HEIGHT;

    setBackground (Color.black);

    page.setColor (Color.blue);
    x = GAP;

    for (int count = 0; count < NUM_BARS; count++)
    {
      height = (int) (Math.random() * MAX_HEIGHT);
      page.fillRect (x, MAX_HEIGHT-height, BAR_WIDTH, height);

      // Keep track of the tallest and shortest bars
      if (height > tallest)
      {
        tallX = x;
        tallest = height;
      }

      if (height < shortest)
      {
        shortX = x;
        shortest = height;
      }

      x = x + BAR_WIDTH + GAP;
    }

    // Redraw the tallest bar in red
    page.setColor (Color.red);
    page.fillRect (tallX, MAX_HEIGHT-tallest, BAR_WIDTH, tallest);

    // Redraw the shortest bar in yellow
    page.setColor (Color.yellow);
    page.fillRect (shortX, MAX_HEIGHT-shortest, BAR_WIDTH, shortest);
  }
}
```

```
<! BarHeights.html>
<HTML>

<HEAD>
<TITLE>The Bar Heights Applet</TITLE>
</HEAD>

<BODY>
<center>

<H3>The BarHeights Applet:</H3>
<APPLET CODE="BarHeights.class" WIDTH=400 HEIGHT=300></APPLET>
<HR>
</center>

</BODY>

</HTML>
```