Chapter 8 Sample Programs

```
//********************************************************************
//  Zero.java       Author: Lewis and Loftus
//  Demonstrates an uncaught exception.
//********************************************************************
public class Zero
{
   //-----------------------------------------------------------------
   //  Deliberately divides by zero to produce an exception.
   //-----------------------------------------------------------------
   public static void main (String[] args)
   {
      int numerator = 10;
      int denominator = 0;

      System.out.println (numerator / denominator);

      System.out.println ("This text will not be printed.");
   }
}

//********************************************************************
//  ProductCodes.java       Author: Lewis and Loftus
//  Demonstrates the use of a try-catch block.
//********************************************************************
import cs1.Keyboard;

public class ProductCodes
{
   //-----------------------------------------------------------------
   //  Counts the number of product codes that are entered with a
   //  zone of R and and district greater than 2000.
   //-----------------------------------------------------------------
   public static void main (String[] args)
   {
      String code;
      char zone;
      int district, valid = 0, banned = 0;

      System.out.print ("Enter product code (XXX to quit): ");
      code = Keyboard.readString();

      while (!code.equals ("XXX"))
      {
         try
         {
            zone = code.charAt(9);
            district = Integer.parseInt(code.substring(3, 7));
            valid++;
            if (zone == 'R' && district > 2000)
               banned++;
         }
```

```
        catch (StringIndexOutOfBoundsException exception)
        {
          System.out.println ("Improper code length: " + code);
        }
        catch (NumberFormatException exception)
        {
          System.out.println ("District is not numeric: " + code);
        }

        System.out.print ("Enter product code (XXX to quit): ");
        code = Keyboard.readString();
      }

      System.out.println ("# of valid codes entered: " + valid);
      System.out.println ("# of banned codes entered: " + banned);
    }
}


//********************************************************************
//  Propagation.java       Author: Lewis and Loftus
//  Demonstrates exception propagation.
//********************************************************************
public class Propagation
{
   //-----------------------------------------------------------------
   //  Invokes the level1 method to begin the exception demonstation.
   //-----------------------------------------------------------------
   static public void main (String[] args)
   {
      ExceptionScope demo = new ExceptionScope();

      System.out.println("Program beginning.");
      demo.level1();
      System.out.println("Program ending.");
   }
}
```

```
//********************************************************************
//  ExceptionScope.java       Author: Lewis and Loftus
//  Demonstrates exception propagation.
//********************************************************************
public class ExceptionScope
{
  //-----------------------------------------------------------------
  //  Catches and handles the exception that is thrown in level3.
  //-----------------------------------------------------------------
  public void level1()
  {
    System.out.println("Level 1 beginning.");

    try
    {
      level2();
    }
    catch (ArithmeticException problem)
    {
      System.out.println ();
      System.out.println ("The exception message is: " +
                          problem.getMessage());
      System.out.println ();
      System.out.println ("The call stack trace:");
      problem.printStackTrace();
      System.out.println ();
    }

    System.out.println("Level 1 ending.");
  }

  //-----------------------------------------------------------------
  //  Serves as an intermediate level.  The exception propagates
  //  through this method back to level1.
  //-----------------------------------------------------------------
  public void level2()
  {
    System.out.println("Level 2 beginning.");
    level3 ();
    System.out.println("Level 2 ending.");
  }

  //-----------------------------------------------------------------
  //  Performs a calculation to produce an exception.  It is not
  //  caught and handled at this level.
  //-----------------------------------------------------------------
  public void level3 ()
  {
    int numerator = 10, denominator = 0;

    System.out.println("Level 3 beginning.");
    int result = numerator / denominator;
    System.out.println("Level 3 ending.");
  }
}
```

```
//********************************************************************
//  CreatingExceptions.java       Author: Lewis and Loftus
//  Demonstrates the ability to define an exception via inheritance.
//********************************************************************

import OutOfRangeException;
import cs1.Keyboard;

public class CreatingExceptions
{
   //-----------------------------------------------------------------
   //  Creates an exception object and possibly throws it.
   //-----------------------------------------------------------------
   public static void main (String[] args) throws OutOfRangeException
   {
      final int MIN = 25, MAX = 40;

      OutOfRangeException problem =
         new OutOfRangeException ("Input value is out of range.");

      System.out.print ("Enter an integer value between " + MIN +
                        " and " + MAX + ", inclusive: ");
      int value = Keyboard.readInt();

      //  Determines if the exception should be thrown
      if (value < MIN || value > MAX)
         throw problem;

      System.out.println ("End of main method.");  // may never reach
   }
}

//********************************************************************
//  OutOfRangeException.java       Author: Lewis and Loftus
//  Represents an exceptional condition in which a value is out of
//  some particular range.
//********************************************************************

public class OutOfRangeException extends Exception
{
   //-----------------------------------------------------------------
   //  Sets up the exception object with a particular message.
   //-----------------------------------------------------------------
   OutOfRangeException (String message)
   {
      super (message);
   }
}
```

```
//********************************************************************
//  Inventory.java       Author: Lewis and Loftus
//  Demonstrates the use of a character file input stream.
//********************************************************************
import InventoryItem;
import java.util.StringTokenizer;
import java.io.*;
public class Inventory
{
   //-----------------------------------------------------------------
   //  Reads data about a store inventory from an input file,
   //  creating an array of InventoryItem objects, then prints them.
   //-----------------------------------------------------------------
   public static void main (String[] args)
   {
      final int MAX = 100;
      InventoryItem[] items = new InventoryItem[MAX];
      StringTokenizer tokenizer;
      String line, name, file="inventory.dat";
      int units, count = 0;
      float price;

      try
      {
         FileReader fr = new FileReader (file);
         BufferedReader inFile = new BufferedReader (fr);

         line = inFile.readLine();
         while (line != null)
         {
            tokenizer = new StringTokenizer (line);
            name = tokenizer.nextToken();
            try
            {
               units = Integer.parseInt (tokenizer.nextToken());
               price = Float.parseFloat (tokenizer.nextToken());
               items[count++] = new InventoryItem (name, units, price);
            }
            catch (NumberFormatException exception)
            {  System.out.println ("Error in input. Line ignored:");
               System.out.println (line);
            }
            line = inFile.readLine();
         }
         inFile.close();
         for (int scan = 0; scan < count; scan++)
            System.out.println (items[scan]);
      }
      catch (FileNotFoundException exception)
      {     System.out.println ("The file " + file + " was not found.");
      }
      catch (IOException exception)
      {     System.out.println (exception);
      }
   }
}
```

```
//********************************************************************
//  InventoryItem.java       Author: Lewis and Loftus
//  Represents an item in the inventory.
//********************************************************************
import java.text.DecimalFormat;

public class InventoryItem
{
   private String name;
   private int units;    // number of available units of this item
   private float price;  // price per unit of this item
   private DecimalFormat fmt;

   //-----------------------------------------------------------------
   //  Sets up this item with the specified information.
   //-----------------------------------------------------------------
   public InventoryItem (String itemName, int numUnits, float cost)
   {
      name = itemName;
      units = numUnits;
      price = cost;
      fmt = new DecimalFormat ("0.##");
   }

   //-----------------------------------------------------------------
   //  Returns information about this item as a string.
   //-----------------------------------------------------------------
   public String toString()
   {
      return name + ":\t" + units + " at " + price + " = " +
            fmt.format ((units * price));
   }
}

//********************************************************************
//  TestData.java       Author: Lewis and Loftus
//  Demonstrates the use of a character file output stream.
//********************************************************************
import java.io.*;
public class TestData
{

   //-----------------------------------------------------------------
   //  Creates a file of test data that consists of ten lines each
   //  containing ten integer values in the range 0 to 99.
   //-----------------------------------------------------------------
   public static void main (String[] args) throws IOException
   {
      final int MAX = 10;

      int value;
      String file = "test.dat";

      FileWriter fw = new FileWriter (file);
      BufferedWriter bw = new BufferedWriter (fw);
      PrintWriter outFile = new PrintWriter (bw);
```

```
      for (int line=1; line <= MAX; line++)
      {
        for (int num=1; num <= MAX; num++)
        {
          value = (int) (Math.random() * 100);
          outFile.print (value + "   ");
        }
        outFile.println ();
      }

      outFile.close();
      System.out.println ("Output file has been created: " + file);
    }
}
```