CSE 341 — Assignment 3
October 27, 1998 — Solution

Write and test a Java program that accepts a URL from the command line and prints out a list of all the links from that page. For example, here is a sample output from my program:

```
orcas% java WebWorm http://www.cs.washington.edu/education/courses/341/CurrentQtr/

Links found in http://www.cs.washington.edu/education/courses/341/CurrentQtr/
 href="mailto:borning@cs.washington.edu"
 href="http://www.cs.washington.edu/homes/hartline/"
 href="mailto:hartline@cs.washington.edu"
 href="general/overview.html"
 href="general/schedule.html"
 href="http://www.cs.washington.edu/education/syllabi/341.html"
 href="general/families.html"
 href="scheme/"
 href="java/"
 href="assignments/"
 href="http://www.cs.washington.edu/education/courses/341/98sp/index.html"
 href="http://www.cs.washington.edu/education/courses/341/98wi/index.html"
 href="http://www.cs.washington.edu/education/courses/341/97au/index.html"
```

Solution:

```
/*
 * WebWorm Solution.   CSE 341.
 *
 * Jason Hartline
 */

import java.net.*;
import java.io.*;
import java.util.*;


/*
 * this class reads in web pages and will, when printed, display
 * the links in the web page.
 */
class WebWorm {

   /*
```

```
 * this holds the links that we have found thus far.
 */
Vector links;

/*
 * this is the URL that we are going to look for links on
 */
String url;

/*
 * sequences of characters denotes a link.
 */
static final String linkPrefix = "<a";
static final String linkSuffix = ">";

/*
 * This contstructor makes a WebWorm specifing the URL to open.
 * It does not actually try to open the URL or verify that it is
 * valid.
 */
public WebWorm(String urlIn)
{
   url = urlIn;
   links = null;
}

/*
 * this function reads in the URL
 * and processes the URL filling up
 * 'links' with all the links found.
 */
public boolean process()
{

   /*
    * prepare to process
    */
   clearError();
   links = null;


   /*
    * try to open our connection to the web server
    * with the URL.
```

```
 */
DataInputStream in = null;

try
{
   URL address = new URL(url);
   in = new DataInputStream(address.openStream());
}
catch (IOException e)
{
   setError("Could not open URL connection.");
   return false;
}



/*
 * initialize our list of hyperlinks.
 */
links = new Vector();
String line = null;

while (true) // break when line = null.
{
   /*
    * try to read in a line.
    */
   try
   {
      line = in.readLine();
   }
   catch (IOException e)
   {
      links = null;
      setError("Error reading from URL connection.");

      /*
       * close the file.
       */
      try
      {
         in.close();
      }
      catch (IOException e2)
```

```
         {
            // do nothing.  We are already dealing with an error.
         }

         return false;
      }

      /*
       * we are done if line == null.
       */
      if (line == null)
         break;

      /*
       * process the line.
       */
      processLine(line);
   }


   /*
    * close the URL connection.
    */
   try
   {
      in.close();
   }
   catch (IOException e)
   {
      links = null;
      setError("Could not close URL connection.");
      return false;
   }

   /*
    * success!!!
    */
   return true;
}

/*
 * This function is a helper function
 * that takes a line and adds any URLs
 * found in that line, to 'links'.
```

```
 */
private void processLine(String s)
{
   /*
    * use a lowercase version of the string to be
    * case insensitive.
    */
   String lower = s.toLowerCase();
   int index=0;

   /*
    * for each link...
    */
   for (index=lower.indexOf(linkPrefix,0);
        index >= 0;
        index = lower.indexOf(linkPrefix,index+1))
   {
      /*
       * find the end of the link.
       */
      int endIndex = lower.indexOf(linkSuffix, index);

      if (endIndex < 0)
         endIndex = s.length();

      /*
       * add the link to 'links'
       */
      links.addElement(s.substring(index+2,endIndex));

      /*
       * prepare to find the next link.
       * on this line.
       */
      index = endIndex;
   }
}

/*
 * printing out a WebWorm will either.
 *    1. if the URL has not been processed, print out
 *       that the WebWorm will process the link.
 *    2. if there was an error processing the URL,
 *       print out the error message.
```

```
 *     3. if the URL was successfully processed, print
 *        out all of the links.
 */
public String toString()
{
   /*
    * oops! there was an error.
    */
   if (hasError())
      return("WebWorm error: " + getErrorMessage());

   /*
    * URL has not been processed yet.
    */
   if (links == null)
      return("WebWorm to load: " + url);

   /*
    * print out header.
    */
   StringBuffer out = new StringBuffer("Links found in " + url);

   /*
    * print out links.
    */
   for(Enumeration e = links.elements(); e.hasMoreElements(); )
      out.append("\n"+e.nextElement());

   return(out.toString());
}

/*
 * main program.
 *
 * - create a WebWorm.
 * - have it process the URL.
 * - print it out.
 */
public static void main(String [] args)
{
   if (args.length != 1)
   {
      System.err.println("USAGE: WebWorm url");
      System.exit(-1);
```

```
        }

        WebWorm worm = new WebWorm(args[0]);

        worm.process();

        System.out.println(worm);
    }


    /*
     * dealing with errors.
     *
     * setError, clearError, hasError, and getError
     */

    private String errorMessage = null;

    private void setError(String error)
    {
        errorMessage = error;
    }

    private void clearError()
    {
        errorMessage = null;
    }

    public boolean hasError()
    {
        return (errorMessage != null);
    }

    public String getErrorMessage()
    {
        return errorMessage;
    }

}
```

Output:

```
Links found in http://www.cs.washington.edu/education/courses/cse341/CurrentQtr/
 href="http://www.cs.washington.edu/homes/borning/"
```

```
href="mailto:borning@cs.washington.edu"
href="http://www.cs.washington.edu/homes/hartline/"
href="mailto:hartline@cs.washington.edu"
href="general/overview.html"
href="general/schedule.html"
href="http://www.cs.washington.edu/education/syllabi/341.html"
HREF=hype/
HREF="general/mailinglist.html"
HREF="general/books.html"
HREF="section/"
href="general/families.html"
href="scheme/"
href="java/"
href="assignments/"
href="http://www.cs.washington.edu/education/courses/341/98sp/index.html"
href="http://www.cs.washington.edu/education/courses/341/98wi/index.html"
href="http://www.cs.washington.edu/education/courses/341/97au/index.html"
```