CSE 341: Homework 4

due 10/25/99

You are to implement all required programs and classes in Java and run them in the Java Runtime Environment. You are to turn in a hardcopy of your source code as well as output of your code on a demonstrative set of test cases.

The topics covered in this assignment include: writing Java code, compiling Java, running Java, using the java class library, and handling exceptions.

Part I: (Not to be turned in.)

Try these out before section.

1. What class should you use to download a URL? Give the full class name including the package information. For example `Object` is really `java.lang.Object`. How would you download the URL?
2. What classes could you use to keep an unsorted set of objects?
3. What classes could you use to read numbers from a file?

Part II: (Not to be turned in)

This will be discussed during section.

1. Write a program that takes in its command line arguments and, as numbers, adds them up and prints out the result.

Part III: (Due Monday, 10/25/99. Turn this in.)

1. (5 points)

   Write a class `HTMLWordParser` that has a constructor that takes an `InputStream` as its only argument. It should have at least the method `nextWord()` that returns the next word in the file, `null` if there are none. It should attempt to discard HTML tags of the form `<text in here>`. For example:

   ```
   <HTML>
   <HEAD>
   <TITLE>My Web Page</TITLE>
   </HEAD>
   <BODY>
   Try this <A HREF=link.html>link</A>
   </BODY>
   </HTML>
   ```

   This HTML page has words: `My`, `Web`, `Page`, `Try`, `this`, and `link`.

   Write a simple Java program to test this class.

2. (10 points)

   Write a program `WebSearch` that takes a list of URLs from the command line and allows the user to search these URLs for some given text. Until the user types `quit`, the program should prompt the user to enter a word to search for and then it should output, on separate lines, the URLs of all the web pages that contain the word. If the user types multiple words, it should print the URLs of only the pages that contain all of the words.

   This program should be robust in that it makes its best effort to do the correct thing despite many possible sources for errors. For example, some URLs might not exist and some URLs might contain have malformed HTML. It is acceptable to give the user warnings when errors occur, but it should not crash or quit prematurely.

   Your program should work as follows:

```
fiji% java WebSearch http://www.yahoo.com/ http://www.washington.edu/
WebSearch> the
http://www.yahoo.com/
http://www.washington.edu/

WebSearch> rainier
http://www.washington.edu/

WebSearch> the rainier
http://www.washington.edu/
```

Part IV: (Assignment 5 Preparation)

Assignment 5 will be to write a cool java applet or stand-alone application, of your choosing. You may do your projects individually or in groups of two. Proportionately more work will be expected from groups than individuals. A typical project should take advantage of at least one of several special facilities provided by Java: the AWT, threads, networking, RMI, etc. This is not an explicit requirement though. A project that does not use one of these special features should be exceptional for other reasons. You should use good, object-oriented style in your program. This will be considered when your project is evaluated. The code should be primarily your own. If you base your code on existing code (there are tons of applets out there), make sure you document this in your project report.

Here are a few suggestions for projects – you can also do something quite different of your own choosing.

- A WWW site navigator. This would display the links and allow navigation. You don't need to produce Netscape Navigator or Internet Explorer – but some simpler functionality should be quite possible.
- An arcade style game, either as an applet or a stand-alone application. An applet might be more fun because then you could put it on your web page.

Various of the games in the Budd Java text could form the basis for this, or you could pick another game.

- A multi-player game, allowing internet play.
- An algorithm animation applet - maybe one that brings your favorite CSE326 data structures to life. Or a hardware animation applet for 378 fans.
- Neato projects from previous years include:
  - A pinball game that read level configuration from the web. It dynamically loaded new classes from the web as well.
  - The `vi` editor.
  - A scheme interpreter.
  - An implementation of shortest-paths algorithm for the UW campus map.

While working on Assignment 4, you should be thinking of a project that you might want to do for Assignment 5. Proposals will be due Wednesday, 10/27/99.