

```
1 import components.sequence.Sequence;
2 import components.sequence.Sequence1L;
3 import components.simplewriter.SimpleWriter;
4 import components.simplewriter.SimpleWriter1L;
5 import components.stack.Stack;
6 import components.stack.Stack1L;
7
8 /**
9  * HW20
10 *
11 * @author Sam Espanioly
12 */
13 public final class practice {
14
15     /**
16      * Default constructor--private to prevent
17      * instantiation.
18      */
19     private practice() {
20         // no code needed here
21     }
22
23     /**
24      * Reverses ("flips") {@code this}.
25      *
26      * @updates this
27      * @ensures this = rev(#this)
28      */
29     //this method works though its no an instance method
30     public void flipSequenceRecursive() {
31         // temporary stack
32         Sequence<String> tempo = new Sequence1L<>();
33
34         if (this.length() > 0) {
```

```
34         tempo.add(0, this.remove(0));
35         this.flipSequenceRecursive();
36     }
37     this.transferFrom(tempo);
38 }
39
40 // this method should work but it gives an error
41 public void flipStack() {
42     // temporary stack
43     Stack<String> tempo = new Stack1L<>();
44
45     for (String x : this) {
46         tempo.push(x);
47     }
48     int len = this.length();
49     for (int i = 0; i < len; i++) {
50         this.transferFrom(tempo);
51     }
52 }
53
54 /**
55  * Reverses ("flips") {@code this}.
56  *
57  * @updates this
58  * @ensures this = rev(#this)
59  */
60 public void flipSequence() {
61     Sequence<String> tempis = new Sequence1L<>();
62     int a = 0;
63     for (String x : this) {
64         tempis.add(a, x);
65         a++;
66     }
67     int len = this.length();
```

```
68         for (int i = 0; i < len; i++) {
69             this.transferFrom(tempis);
70         }
71     }
72
73     /**
74      * Main method.
75      *
76      * @param args
77      *         the command line arguments; unused here
78      */
79     public static void main(String[] args) {
80         SimpleWriter out = new SimpleWriter1L();
81     }
82
83 }
84
```