OSU CSE 2221 – Software 1: Software Components

Lecturer: Nyigel Spann

Project #9: String Reassembly From Fragments

The Ohio State University

College of Engineering

Columbus, Ohio

```
import static org.junit.Assert.assertEquals;

import org.junit.Test;

import components.set.Set;
import components.set.Set1L;
import components.simplereader.SimpleReader;
import components.simplereader.SimpleReader1L;
import components.simplewriter.SimpleWriter;
import components.simplewriter.SimpleWriter1L;

/**
 *
 * @author Danny Kan (kan.74@osu.edu)
 *
 */
public class StringReassemblyTest {

    /*
     * Tests of combination
     */

    @Test
    public void testCombination1() {
        String myStr1 = "abcdef";
        String myStr2 = "defghi";
        int overlap = 3;
        String expected = StringReassembly.combination(myStr1, myStr2, overlap);
        assertEquals("abcdefghi", expected);
```

```
}

@Test
public void testCombination2() {
   String myStr1 = "computer_science";
   String myStr2 = "_science_and_engineering";
   int overlap = 8;
   String expected = StringReassembly.combination(myStr1, myStr2, overlap);
   assertEquals("computer_science_and_engineering", expected);
}

@Test
public void testCombination3() {
   String myStr1 = "123";
   String myStr2 = "3456";
   int overlap = 1;
   String expected = StringReassembly.combination(myStr1, myStr2, overlap);
   assertEquals("123456", expected);
}

/*
 * Tests of addToSetAvoidingSubstrings
 */

@Test
public void testAddToSetAvoidingSubstrings1() {
   Set<String> mySet = new Set1L<>();
   mySet.add(" abc def ");
   mySet.add("ghi");
```

```
    mySet.add("jkl");
    String myStr = "xyz";
    Set<String> expected = new Set1L<>();
    expected.add(" abc def ");
    expected.add("ghi");
    expected.add("jkl");
    expected.add("xyz");
    StringReassembly.addToSetAvoidingSubstrings(mySet, myStr);
    assertEquals(mySet, expected);
}

@Test
public void testAddToSetAvoidingSubstrings2() {
    Set<String> mySet = new Set1L<>();
    mySet.add("something");
    mySet.add("abc");
    mySet.add("123");
    String myStr = "xyz";
    Set<String> expected = new Set1L<>();
    expected.add("something");
    expected.add("abc");
    expected.add("123");
    expected.add("xyz");
    StringReassembly.addToSetAvoidingSubstrings(mySet, myStr);
    assertEquals(mySet, expected);
}

@Test
public void testAddToSetAvoidingSubstrings3() {
```

```
    Set<String> mySet = new Set1L<>();
    mySet.add("danny");
    mySet.add("dan");
    mySet.add("kan");
    String myStr = "x";
    Set<String> expected = new Set1L<>();
    expected.add("danny");
    expected.add("dan");
    expected.add("kan");
    expected.add("x");
    StringReassembly.addToSetAvoidingSubstrings(mySet, myStr);
    assertEquals(mySet, expected);
}

/*
 * Tests of linesFromInput
 */

@Test
public void testLinesFromInput1() {
    SimpleReader input = new SimpleReader1L("testLinesFromInput1.txt");
    Set<String> mySet = StringReassembly.linesFromInput(input);
    Set<String> expected = new Set1L<>();
    expected.add("a");
    expected.add("b");
    expected.add("c");
    assertEquals(mySet, expected);
    input.close();
}
```

```
@Test
public void testLinesFromInput2() {
   SimpleReader input = new SimpleReader1L("testLinesFromInput2.txt");
   Set<String> mySet = StringReassembly.linesFromInput(input);
   Set<String> expected = new Set1L<>();
   expected.add("abc");
   expected.add("danny");
   expected.add("abc");
   assertEquals(mySet, expected);
   input.close();
}

@Test
public void testLinesFromInput3() {
   SimpleReader input = new SimpleReader1L("testLinesFromInput3.txt");
   Set<String> mySet = StringReassembly.linesFromInput(input);
   Set<String> expected = new Set1L<>();
   expected.add("   qwerty");
   expected.add("random    ");
   expected.add("foo");
   assertEquals(mySet, expected);
   input.close();
}

/*
 * Tests of printWithLineSeparators
 */
```

```
@Test
public void testPrintWithLineSeparators1() {
   String myStr = "this~ is~ a~ test";
   SimpleWriter output = new SimpleWriter1L("testWithLineSeparators1.txt");
   String expected = "this is a test";
   StringReassembly.printWithLineSeparators(myStr, output);
   assertEquals("this is a test", expected);
   output.close();
}

@Test
public void testPrintWithLineSeparators2() {
   String myStr = "   xyz~xyz~xyz   ";
   SimpleWriter output = new SimpleWriter1L("testWithLineSeparators2.txt");
   String expected = "   xyzxyzxyz   ";
   StringReassembly.printWithLineSeparators(myStr, output);
   assertEquals("   xyzxyzxyz   ", expected);
   output.close();
}

@Test
public void testPrintWithLineSeparators3() {
   String myStr = "~~~x~y~z~_~_~something ~ goes ~ here";
   SimpleWriter output = new SimpleWriter1L("testWithLineSeparators3.txt");
   String expected = "xyz__something  goes  here";
   StringReassembly.printWithLineSeparators(myStr, output);
   assertEquals("xyz__something  goes  here", expected);
   output.close();
}
```

```
}
```