COMP30820 Java Programming (Conv)

Programming Test

Time Allowed: 70 minutes

1 Instructions

- Download file `test.zip` from Moodle. Unzip the file and copy the package into Eclipse.

- Rename the **package** as follows: `test_12345678` where `12345678` is your student number.

- Write your methods in the classes provided (one method per class). Each class contains a `main` method with some test cases included to enable you to test your code:
    - *Include your* **name** *and* **student number** *as a comment at the beginning of each class.*

- At the end of the test, upload your solutions to Moodle:
    - Create a new folder (e.g. on your desktop) using the name `test_12345678` where `12345678` is your student number. *The name of this folder and the name of your Eclipse package must be the* **same**.
    - Copy the **.java** classes from your Eclipse package into the folder.
    - Zip the folder and upload the zip file to Moodle using the link in section: `Assessment - Programming Test`.

- Please note:
    - This is an open book test. You may refer to the lecture notes, the textbook, and any notes you may have prepared yourself. Access to the Web is **not** permitted except for the Java API page and the course Moodle page.
    - **Important** — this is **not** a team/group exercise — each student must submit her/his own work. Please ask if you have any questions about this. See the course Moodle for information on the UCD plagiarism policy.

2 Marking Scheme

For each question, the following marking scheme will be used:

- 1 mark for each correct test case. Some test cases are given to you in the code you have downloaded from Moodle; a number of additional test cases will also be used to grade your solution.

3 Questions

Answer all questions.

Question 1

Write a method to return the sum of all digits (`'0'` to `'9'`, inclusive) in a string. Use the following method header:

```
public static int getSumDigits(String str)
```

Test cases:

- For input string `""`, the method should return: 0
- For input string `"hello"`, the method should return: 0
- For input string `"a2b3"`, the method should return: 5

Write this method in file `Q1.java`.

Question 2

Write a method to return the Largest Common Suffix (LCS) of two strings. Use the following method header:

```
public static String getLCS(String s1, String s2)
```

Test cases:

- For input strings `"Hello"` and `"HELLO"`, the method should return the empty string: `""` (i.e. a string which contains no characters)
- For input strings `"computing"` and `"working"`, the method should return: `"ing"`

Write this method in file `Q2.java`.

Question 3

A string of parentheses is said to be correctly balanced if each opening parenthesis has a corresponding closing parenthesis and pairs of parentheses are properly nested.

Test cases:

– The following strings of parentheses **are** correctly balanced:

```
""
"()"
"()()"
"(())"
```

– The following strings of parentheses **are not** correctly balanced:

```
"("
")"
"(()"
"()(("
```

Write a method that returns `true` if a string of parentheses is correctly balanced, and `false` otherwise. Assume strings contain only parentheses characters. Use the following method header:

```java
public static boolean isBalanced(String p)
```

Write this method in file `Q3.java`.