

SEMESTER 1, 2020 SAMPLE EXAM

CITS5501 Software Testing and Quality Assurance

This paper contains: 7 pages (including title page)

Time allowed: 135 minutes (including reading time)

Instructions:

- The exam has 5 questions with a total value of 50 marks.
- The time allowed to complete the exam is 135 minutes. (2 hour exam plus 15 minutes to allow for any technology overheads)
- This is an ExamSoft exam.
- Printed or handwritten notes are permitted.
- No electronic devices are permitted during the examination.
- Calculators are not allowed for this exam.
- Students may use blank paper for rough working during the exam.
- All answers must be typed in ExamSoft for marking.
- Any feedback for the examiner must be entered using the ExamSoft Notes feedback feature.

replace this page with official cover page 1

replace this 2nd page with official cover page 2  
(that is, “this page has been left blank”)

## wordCount method

Questions 1 and 2 refer to the following Java wordCount method.

```
1  /** Returns the number of unique words in the string "aString"
2   * (where a "word" is a contiguous sequence of non-whitespace
3   * characters).
4   * If "aString" is null, the result is undefined.
5   */
6  public static int wordCount(String aString) {
7      aString = aString.trim(); // trim whitespace from either end
8
9      // handle case where string has no words
10     if (aString.length() == 0) {
11         return 0;
12     }
13
14     String words[] = aString.split("\\s+");
15
16     ArrayList<String> uniqueWords = new ArrayList<>();
17     int count = 0;
18     for (int i = 0; i < words.length; i++) {
19         String w = words[i];
20         if (! uniqueWords.contains(w) ) {
21             count += 1;
22             uniqueWords.add(w);
23         }
24     }
25     return count;
26 }
```

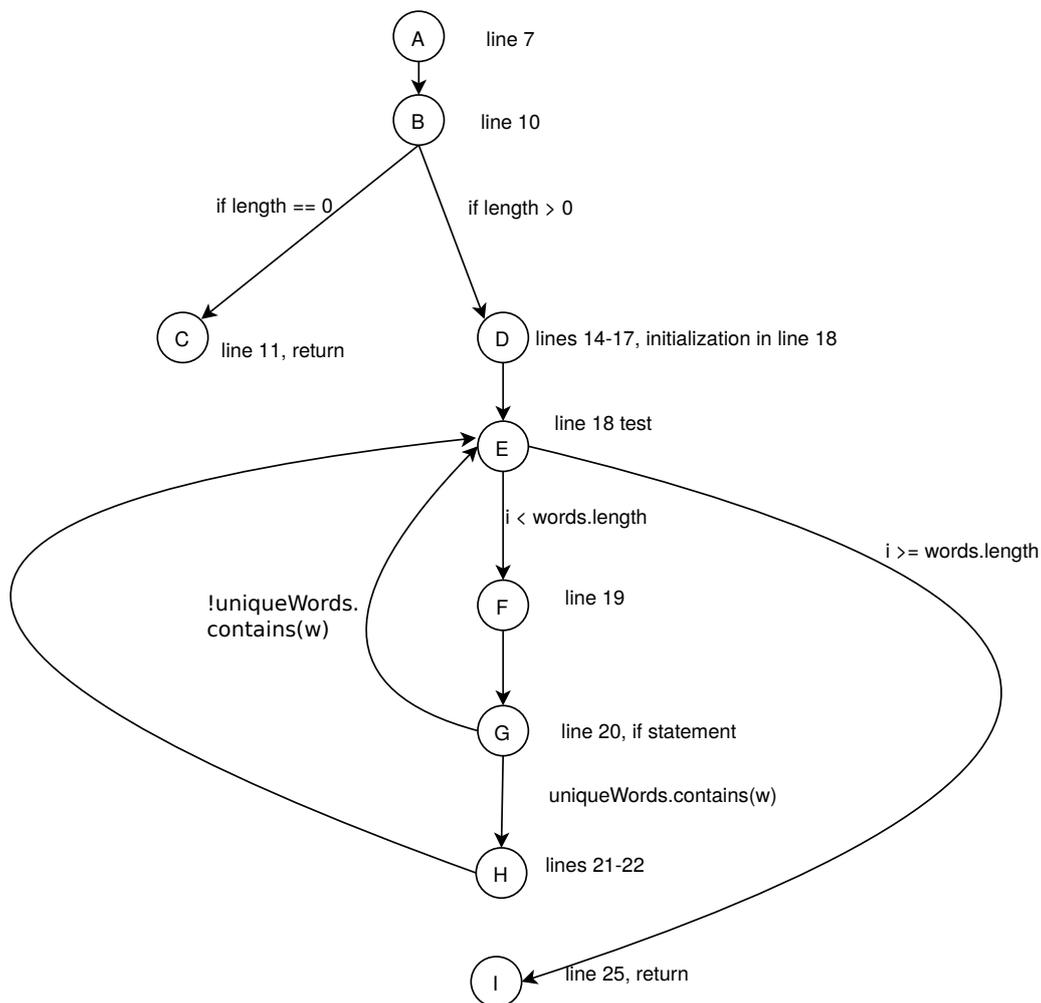
1. Consider the wordCount method described above.

Using input space partitioning, describe some partitions from which test cases for this function could be drawn, showing your working. Describe three unit tests derived from those partitions. You need not present code for the unit tests, but should clearly describe the input and expected results.

[10 marks]

2. Consider the Java `wordCount()` method described previously.

The following diagram is intended to represent the flow of control through the function.



- i. Explain what a *simple path* and a *prime path* are.
- ii. Identify three simple paths through the diagram that are *not* prime paths, justifying your answer.
- iii. Identify two prime paths through the diagram, justifying your answer.
- iv. Identify any simplifying assumptions which have been made in producing the diagram.

[10 marks]

3. You are on the software quality assurance team for Rook Capital, which specialises in executing trades for stock and futures brokers. The software development team is working on a program, “UBAR” (“Unbelievably Brisk Automatic Ratiocinator”) for automatically executing trades. It takes as input sets of rules (effectively, programs described in a simple language) supplied by Rook Capital’s clients, monitors stock prices, and automatically executes trades based on the rules. An incorrectly executed trade could result in enormous losses for clients, so it is important the system be highly reliable: it should function correctly, transmit data securely, and have minimal downtime.

How would you develop a *risk management* plan for Rook Capital? Explain the steps involved. Identify at least three risks, assess their impact and chance of occurring, and explain how you would mitigate them.

[10 marks]

4. You are designing tests for a back-end banking system, which takes as input a list of transactions to perform.

The input format is specified by the following grammar:

```
<transaction> ::= <deposit> | <withdrawal>
<deposit>      ::= "DEP" <accountId> <amount>
<withdrawal>   ::= "WDR" <accountId> <amount>
<amount>       ::= "." <digit> <digit> | <digit><amount>
<accountId>    ::= <digit><digit><digit> "-" <digit><digit><digit>
<digit>        ::= "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"
```

- i. Explain how *terminal coverage* and *production coverage* differ.
- ii. If you were writing syntax-based tests for the system, based on the grammar above, how many tests would be required to achieve terminal coverage? And how many for production coverage? Explain your answer.
- iii. Provide one test based on the grammar: a description of the test, the inputs, and the expected output. If you need to make any assumptions, state what they are.

[10 marks]

5. You are part of a team developing an online, searchable movie database called “YAMDB” (“Yet Another Movie Database”), and you are using the Alloy analyzer to develop a formal model of your system.

Your team has identified the following entities, relationships and constraints which your system should represent:

- The system models relationships between entities called *Movies*.
- A movie may have one movie which is a direct *sequel* to it. (Or, it may not have a sequel.)
- A movie cannot be its own sequel. (As a result of this, certain genres of time-travel film are outside the scope of YAMDB.)
- If we have two movies,  $a$  and  $b$ , then it cannot be the case that they are both each other’s sequel.
- In addition to its direct sequel, a movie has a set (possibly empty) of *sequels*, defined as follows:
  - If movie  $b$  is the direct sequel of movie  $a$ , then  $b$  is in the sequels of  $a$ .
  - If movie  $m$  is in the sequels of movie  $a$ , and movie  $n$  is the direct sequel of movie  $m$ , then  $n$  is in the sequels of  $a$ .
  - No other movies are sequels of  $a$ .

Give Alloy definitions which model a “Movie” entity, with the constraints described above.

[10 marks]

END OF PAPER