UNIVERSITY OF EDINBURGH

COLLEGE OF SCIENCE AND ENGINEERING

SCHOOL OF INFORMATICS

SOFTWARE TESTING

Friday 16 May 2008

14:30 to 16:30

Year 3 Courses

Convener: G Hayes
External Examiners: J Gurd, M Wooldridge

INSTRUCTIONS TO CANDIDATES

Answer any TWO questions.

All questions carry equal weight.

CALCULATORS MAY NOT BE USED IN THIS EXAMINATION

1. The `String.split()` method was added to the Java specification at version 1.4, and is the preferred modern replacement for the legacy `StringTokenizer` class:

   ```
   // class String
   public String[] split(String regex, int limit)
   ```

   `split()` breaks *this* `String` into an array of fragments delimited by matches of the regular expression *regex*. No more than *limit* fragments are returned (the `/*i*/` comments here are shown purely to highlight how the method operates):

   ```
   s = "mississippi"
   s.split("i", 999) -> { "m",  "ss",  "ss",  "pp",  "" }
                            /*i*/  /*i*/  /*i*/  /*i*/
   ```

   Effect of *limit*:

   - When *limit* is positive, if there are more than ($limit - 1$) matches of *regex* in the string (and hence more than *limit* fragments), then the array will contain exactly *limit* elements, and the last element will contain the unsplit excess from the string (s here has the same value as above):

     ```
     s.split("i", 3) -> { "m", "ss", "ssippi" }
     ```

   - When *limit* is negative, there is no limit on the number of fragments returned:

     ```
     s.split("i", -1) -> { "m", "ss", "ss", "pp", "" }
     ```

   - When *limit* is zero, there is also no limit on the number of fragments returned, **but** all empty **trailing** fragments are elided:

     ```
     s.split("i", 0) -> { "m", "ss", "ss", "pp" }
     s.split("iss", 0) -> { "m", "", "ippi" }
     ```

(a) Using just the information provided here, what test methodology would you apply to an implementation of `String.split()`? Justify your answer. What extra information about the implementation would be useful for testing, and how might access to this information affect the testing process? Suggest reasons why this might improve **or impair** the testing process. [*6 marks*]

(b) Work through the test method you have suggested, in order to generate a test specification for `String.split()`. Make sure that you clearly identify all steps involved in the process as you apply them. Do not use the string "mississippi" in your test specification. [*11 marks*]

Note:

- You should assume that the regular expression features of `split()` are to be tested by someone else, so for this question *regex* is a simple string (as in the examples above).
- If the specification appears ambiguous or inadequate, note this, and make (and declare) any assumptions you feel necessary to allow you to continue with your work.

(c) When testing systems with many parameters, we often initially identify an impractically large number of possible tests. Describe, in reasonable detail, the operation of **two** techniques by which the number of tests can be reduced. Make reference to your answer to part (b) or provide examples. [*8 marks*]

2. The Java method `match()` implements a string search algorithm: it returns the (0-based) index of the first occurrence of `String` *needle* in `String` *haystack*, or -1 if *needle* is not a substring of *haystack*.

```
static int match(String haystack, String needle) {
  for(int i = 0; i + needle.length() <= haystack.length(); ++i) {
    int j = 0;
    while(j < needle.length() &&
        haystack.charAt(i + j) == needle.charAt(j))
      ++j;
    if(j == needle.length())
      return i;
  }
  return -1;
}
```

For example, `match("mississippi", "sis")` would return 3.

(a) Construct a control flow graph for `match()`. [*4 marks*]

(b) What does it mean for one test adequacy criterion to subsume another? Define **six** adequacy criteria you could consider when developing a test suite for this method, and describe the relations between them. [*6 marks*]

(c) Choose three adequacy criteria and write **three** test suites for `match()`, such that each suite satisfies a criterion that its predecessors do not, as shown in this table:

	Criterion A	Criterion B	Criterion C
Suite 1	satisfied	not satisfied	not satisfied
Suite 2	don't care	satisfied	not satisfied
Suite 3	don't care	don't care	satisfied

Do not use the string "mississippi" in your tests. [*9 marks*]

[**Note:** In simple code, even a small test may satisfy many criteria, so you may wish to use your answer to part (b) to identify the weakest and strongest criteria that you can practically apply, and use those as criteria **A** and **C**, respectively.]

(d) Write a brief note on **three** of the following topics:

- How "short-circuit" evaluation of compound conditions affects some coverage criteria.
- How decidability and reachability limit our expectations of certain coverage criteria.
- How data flow testing provides a natural starting point for the development of integration testing.
- Some research indicates that stronger coverage criteria do not correlate with better tests. What is your reaction to this?

[*6 marks*]

3. A new open platform smartphone is on the market: the manufacturer welcomes anybody who wishes to write their own user interface and applications for the device. Despite not having worked with embedded systems before, your usability-focused research group has decided to revolutionise the mobile 'phone world by releasing a state-of-the-art open source user interface and application set for this mobile 'phone, demonstrating your latest ideas. The group has found a wealthy benefactor, so the project is well funded and has a large team working on it.

   (a) Suggest a development methodology for this project. Describe your chosen methodology, with particular attention to how testing fits into its lifecycle. Justify your suggestion, contrasting it briefly (perhaps only on one or two points) with **two** other methodologies. [*6 marks*]

   (b) Identify key characteristics of this system, and the kinds of system tests you would use to help verify these characteristics. Describe and justify each of your suggestions, with particular attention to their respective testing goals and methods.

   After careful consideration, you conclude that your team doesn't have the resources to test thoroughly in all of these areas; which type of testing will you drop first, and why? [*8 marks*]

   (c) You are keenly aware of how distressing it is when a device as seemingly simple as a mobile 'phone malfunctions. Describe and justify **two** approaches or measures you could use to reassure yourself that testing and bug-fixing of your software has been thorough and measurably effective. [*8 marks*]

   (d) Your benefactor is concerned that by the time the project comes to fruition, the smartphone you are developing for will be obsolete. What addition (if any) could you make to your testing regime in order to reassure your benefactor that you will be able to migrate quickly and reliably to the next version of the smartphone? Write a summary of your suggestion for the benefactor, giving justifications, benefits and costs. (If you believe that you need do nothing more, then justify this stance). [*3 marks*]