COMPSCI/SFWRENG 3S03 Software Testing
Midterm Test – Winter 2020-2021

Question 1 (5 points)
Testers sometimes need to replace part of the system they are testing with what is often called a test double: a real component is replaced with a "double", which is a simplified version of the real component that is much easier to test. For example, a database, or a transaction processing engine could be replaced by a simple API or interface that produces dummy (simple, plausible) responses to queries or transaction submissions, instead of going to the expense of running a query or processing a transaction. A mock object is a well-known example of a test double. In around a paragraph, comment on the advantages and disadvantages of using test doubles, including any effect they might have on coverage metrics.

Question 2 (5 points)
A test oracle encodes the expected result of a test. They are essential in automated testing, and are used when the result of running a test needs to be checked to see if it is correct or accurate. Describe three characteristics of a good test oracle, and one characteristic of a poor test oracle.

Question 3 (5 points)
Suppose you wanted to test a speech to text processor (like Dragon Anywhere or Amazon Transcribe), which takes a sound file and produces a transcript (a text file) of what was said. How would you test this? Briefly comment on the expense of testing such a system. If testing is really expensive, can you think of any way to reduce the cost? If testing is really cheap, explain why.

Question 4 (10 points)
Consider the following subset of a program, which includes a "try-catch" block. This is made up of two parts: (1) a "try" block that allows you to specify a block of code that will be tested for failures as it executes; and (2) one or more "catch" blocks, which define code that will be executed if a failure arises when executing the "try" block. Each "catch" block specifies a particular type of Exception (failure) that the specific block will handle.

```
try {
      s=br.readLine();  // read a line of text from console
      if(s.length()>96)
            throw new Exception("Too long");
      if(s.length()==0)
            throw new Exception("Too short");
}(catch IOException e){
      e.printStackTrace();
}(catch Exception e){
```

```
            e.getMessage();
        }
        return(s);
```

a) [10 marks] Draw a control flow graph for this program. You can draw your diagram by hand or with a drawing tool.
b) [5 marks] Is it possible to achieve 100% statement coverage? If so, give a small set of tests that will achieve 100% statement coverage. If not, explain why.
c) [5 marks] Is it possible to achieve 100% branch coverage? If so, give a small set of tests that will achieve 100% branch coverage. If not, explain why.