Maynooth University
National University of Ireland Maynooth

SEMESTER 1

Summer 2021 Examination

CS608

Software Testing

Dr.P. Nicholl, Dr J. Timoney, Dr. S. Brown

Time allowed: 3 hours

Answer at least **three** questions

Your mark will be based on your best ***three*** answers

All questions carry equal marks

Instructions

	Yes	No
Log Books Allowed	☐	☑
Formula Tables Allowed	☐	☑
Other Allowed (*enter details)* The course textbook: Essential Elements of Software Testing	☑	☐

General (*enter details)*

You are taking this examination remotely. By submitting your answers, you confirm that this is all your own work. If you have questions during the exam, ask Dr. Brown using Teams/email.

You may refer only to the book, your course notes, the slides, and lab material during the exam. Note that your answers must be in your own words: you will not receive marks for any text or examples copied directly from the course material.

[25 marks]

1. (a) Explain, with an example of your own (not from the book), why *exhaustive testing* is infeasible. [5 marks]

(b) For the method `whatSpeed()` as defined below, identify the natural and specification-based ranges using value lines. Provide a table defining all the input and output partitions. Note: treat below 0 and above 110 as two separate partitions. [5 marks]

whatSpeed

```
public static CpuCooler.Required whatSpeed(int temp,
boolean variableSpeed)
```

Indicates the speed required for a CPU cooling fan.

Parameters:

`temp` - - the current temperature (in degrees C)

`variableSpeed` - - true if the fan has variable speed control

Returns:

ERROR if the cpu temperature is not within 0..110 degrees
If the fan has variable speed control:

- HIGH if the cpu temperature is above 90 degrees
- LOW if the cpu temperature is above 50 degrees (but not above 90)
- OFF if the cpu temperature is not above 50 degrees

If the fan does not have variable speed control:

- HIGH if the cpu temperature is above 90 degrees
- OFF if the cpu temperature is not above 90 degrees

(c) Develop equivalence partition (EP) tests for the method `whatSpeed` () as defined above. You answer should include tables for: the test coverage items (TCIs), selected equivalence values, and the test cases. Complete the TCI table, showing that each test case is covered. Make sure you have no duplicate coverage in your test cases. [15 marks]

[25marks]

2. (a) Explain the difference between *errors of omission* and *errors of commission,* providing an example of your own in your answer (not from the book). With reference to your example, clearly explain the reason why each type of error is more likely to found by either black-box testing or found by white-box testing. [10 marks]

(b) The method Wind.categorise() categorizes winds by their speed as specified here: [10 marks]

categorise

```
public static cs608.Wind.Speeds categorise(int knots)
```

Categorise the wind speed.

Parameters:

`knots` - - the current windspeed

Returns:

ERROR if the cpu temperature is less than 0 knots
Otherwise:

- CALM if the windspeed is less than 1 knot
- LIGHT if the windspeed is 1-3 knots
- BREEZE if the windspeed is 4-27 knots
- GALE if the windspeed is 28-55 knots
- STORM if the windspeed is 56-63 knots
- HURRICANE if the windspeed >= 64 knots

The following code coverage figure (next page) shows the measured coverage of existing black-box tests for the method categorise(). Note that lines 36, 38 and 40 highlighted in red, and lines 35, 37, 39 in yellow.

Develop the **additional** tests for the method required to provide full statement coverage (SC). In you answer, include the conditions required to execute each unexecuted line in your analysis, the completed test coverage items table, and the test cases.

```
24.
25.    public static Speeds categorise(int knots) {
26.
27.        if (knots<0)
28.            return ERROR;
29.        else if (knots<1)
30.            return CALM;
31.        else if (knots<=3)
32.            return LIGHT;
33.        else if (knots<=16)
34.            return BREEZE;
35.        else if (knots<=27)
36.            return BREEZE;
37.        else if (knots<=55)
38.            return GALE;
39.        else if (knots<=63)
40.            return STORM;
41.        else
42.            return HURRICANE;
43.
44.    }
```

(c) Work out the required input parameter values for v and f that ensure execution of lines 6 and 8 in the method inR() shown here: [5 marks]

```
1 public static boolean inR(int v, boolean f) {
2     int lower=7; int upper=16;
3     if (f) upper = 33;
4     v = v - lower; upper = upper - lower;
5     if (v>=0 && v<=upper)
6        return true;
7     else
8        return false;
9 }
```

[25marks]

3. (a) Explain the difference between conventional testing and testing ‘ *in class context* ’. Include simple examples of each (not from the book) and make reference to it in your answer. [10 marks]

(b) Develop OO tests ‘*in class context*’ for the method heating() in class Thermostat as defined below. You are to test for the following three situations (limit1 is always greater than limit2): [15 marks]

(i) With temp>limit 1, no heating
(ii) With limit2<temp<=limit 1, heating with setting 1
(iii) With temp<=limit 2, heating with setting 2

Provide EP test coverage items and test cases in your answer, using an EP value of 16 for limit1 and 10 for limit2.

```
package cs608;

public class Thermostat {
   int limit1, limit2, setting;
   // Set the temperature limit for heating level 1
   public void setLimit1(int temp) {
      limit1 = temp;
   }
   // Set the temperature limit for heating level 2
   public void setLimit2(int temp) {
      limit2 = temp;
   }
   // Getter for heater setting
   public int getSetting() {
      return setting;
   }
   /**
    * Determine if heating is required
    * and set the heater setting (0=off, 1=low, 2=high).
    * @param temp - the current temperature
    * @return true if temp<limit1 or temp<limit2
    */
   public boolean heating(int temp) {
      boolean required=true;
      if (temp<limit2)
         setting = 2;
      else if (temp<limit1)
         setting = 1;
      else {
        setting = 0;
         required = false;
      }
      return required;
   }
}
```

[25marks]

4. (a) Show the TestNG annotation you would use for each of the following five situations, and explain how it works. If there is more than one way to achieve any of these, select just one of the ways. [10 marks]

(i) Selecting individual tests or sets of tests to execute.
(ii) Executing tests in a specific order.
(iii) Testing code that raises an exception.
(iv) Using parameterized tests.
(v) Running a method to setup a class to test before any of the test methods execute.

(b) Develop random tests for the method bid() defined below which returns true only if a bid is greater than the top bid and the auction is not closed. [15 marks]

Use combinational tests as the basis for the random testing, using the Decision Table provided (no error conditions).

In your answer:
(i) Identify the equivalence value criteria for the input parameters.
(ii) Produce a table of the test cases.
(iii) Describe how this resolves the three problem with automated random testing: test oracle, test data, and test completion.
(iv) Outline how the random data values are produced – assume the method generateRandomInt(lower,upper) is available
(v) Outline how the java code prints out test failure results

Decision table for Auction.accept(int bid, int topBid, Boolean closed) where bid>0 and topBid>0.

	Rules			
	1	2	3	4
Causes				
bid>topBid	T	T	F	F
closed	T	F	T	F
Effects				
Return true	F	T	F	F