

CS265 Software Testing

Lab Assessment EXAM 1 (15% CA)

November 2018

Exam duration: 90 minutes

This is an open-book exam. However you are only permitted use the Moodle resources (lecture notes, lab solutions, etc) for CS265 and whatever written notes that you have taken in lectures and labs. No other web pages should be opened. No talking. Under no circumstances can social media, social media message applications, email or other Internet-based communication be used for the duration of the lab exam. Any violation of these conditions will result in a mark of 0%.

Exercise 1: Black Box Testing [8%]

On Moodle you will see a Java class called `CS265_LabExam1_Q1.java`. Inside this class there is a method `public static String speedMonitor(int speed)`

```
* The control board for a roadside sign for speed measurement
* is fed by the following method.
* Speed < 0 .... display INVALID
* Speed between 0 and 60KM/H ... .display THANK YOU
* Speed between 61 and 80 ... display SLOW DOWN
* Otherwise display TOO FAST
```

You are asked to develop a suite of Black Box Tests in TestNG for the method `speedMonitor` by using the following approaches to generate test cases and test data: (1) Equivalence Partitions (2) Boundary Value Analysis (3) Truth Tables / Decision Tables. **YOU MUST DEVELOP EACH INDEPENDENTLY (WITHOUT REUSE) FROM EACH OTHER.** The following marking scheme applies:

Item Required	Marking Scheme (in %)
Equivalence Partition (Partitions and Test Data)	0 (wrong or not relevant), 1 (good attempt), 2 (fully correct)
Boundary Value Analysis (Boundaries and Test Data)	0 (wrong or not relevant), 1 (good attempt), 2 (fully correct)
Truth Table/Decision Table (with Test Data)	0 (wrong or not relevant), 1 (good attempt), 2 (fully correct)
TestNG Java Code	0 (too few tests implemented), 1 (good attempt but missing some tests), 2 (fully correct – the set of tests provided is sufficient)

What do I need to provide?

1. You must provide a table (or tables) for the Equivalence Partitions, Boundary Value Analysis and the Truth Table/Decision in digital format (uploaded to Moodle).
2. You must upload your TestNG test JAVA file to Moodle.

Exercise 2: White Box Testing [3%]

On Moodle you will find a Java class called `CS265_LabExam1_Q2.java`. Inside this class file there is a method `public static String isIPAddress(String ipAddress)` which indicates whether a given string parameter `ipAddress` is a valid IP address.

```
* We define an IP address as follows. It is a computer address.  
* Usually 1- 3 digits followed by a period followed by 1 - 3 digits  
followed by a period  
* followed by 1 - 3 digits followed by a period followed by 1 - 3 digits.  
* The digits can repeat themselves.  
* For the sake of argument  
* 888.888.999.777 is a valid IP address for this question. 123.001.10.7 is  
a valid IP address for this question.  
*  
* IGNORING the case (upper or lower) of the string  
* If the IP address is null, empty or length less than 5 then return  
INVALID  
* If the IP address has the structure 192.(1 - 3 digits).(1 - 3 digits).  
(1 - 3 digits)  
* then return LOCAL.  
* If the IP address is 127.0.0.1 then return LOCAL  
* If the IP address has the structure (1 - 3 digits).(1 - 3 digits).(1 - 3  
digits).(1 - 3 digits) then return EXTERNAL.  
* Any other combination (including those with characters etc) the string  
INVALID is returned.
```

You are required to develop a set of White Box Tests in TestNG by analysing the source code of the `isIPAddress` method. You should take care to exercise all statements in the method. You should use EclEmma to calculate overall statement coverage

Item Required	Marking Scheme (in %)
Full screenshot of the EclEmma of the method.	0 (screenshot not uploaded or very unclear), 1 (screenshot clearly shows coverage as outlined in lectures) . Please check with demonstrators if you are unclear about taking a screenshot
TestNG Java Code	0 (too few tests implemented, low coverage), 1 (good attempt but not 100% coverage), 2 (100% coverage in EclEmma)

What do I need to provide?

1. You must upload a clear, full screen, screenshot of the EclEmma coverage of the `isIPAddress` method within Eclipse that shows the coverage information.
2. You must upload your TestNG Java Code file for this exercise to the Moodle quiz.

Exercise 3: Control Flow Graph (4%)

Consider the following code:

```
1  public double saleRate(int quantity, boolean cashpayment, boolean wholesale)
2      {
3          double discount=0.00;
4          if (quantity <50)
5              if (cashpayment && wholesale)
6                  discount=0.04;
7              else
8                  {
9                      if (cashpayment ||wholesale)
10                         discount=0.02;
11                  }
12          else
13              {
14                  discount=0.04;
15                  if (cashpayment ||wholesale)
16                      discount=discount+0.02;
17              }
18          return discount;
19      }
```

Draw a CFG that fully represents this source code. Think about how many branches and how many paths the graph has.

Complete the question(s) for this exercise as part of the **Moodle quiz**. You must complete these questions in order to collect the point(s) for this exercise.

Important Note: Graphs may be mirrored based on how you drew your graph, as mentioned in class. Please take this into consideration.

You will need to UPLOAD your work to the Moodle Quiz for this lab exam. This will be the only way you will be assessed for the CA for this lab.