CSE 464: Software Quality Assurance and Testing
Midterm I : Practice Questions

Note: These practice questions do not cover all the topics for exam 1. These practice questions will help you to get an idea about the nature of the exam and types of questions you can expect. Make sure to review all the topics in the study guide, review lectures, examples discussed in the class, recitations, lecture review activities and homeworks. **Solutions for this practice exam will be discussed during the exam review on Thursday during the lecture**

Part I : Multiple Choice and Short Answer

Select the best possible answer for each of the question below

i.) Functional testing does not involve…..

a) Exploratory testing
b) Boundary value analysis
c) Source code
d) Black box testing approaches

ii.) Which of the following is a NOT factor that software testers used to design equivalent class based test cases.

A) valid inputs B) invalid inputs C) output D) System configuration
E) Cyclomatic complexity

iii.) Decision table technique should be used in a situation where…..

A) Variables are integers B) Inputs variables are not independent
C) Variables are independent D) code is available to the tester

iv.) If a Java program read an age of a person and the valid range for ages are between 0 and 100 inclusive, which of the following has a correct boundary value for age in robustness scenario

a) -1, 0, 1 54 99 100 101 b) 0 54 100 c) 0 1 54 99 100 d) -1 0 1 99 100 101

v.) Which of the following is NOT true about exploratory testing

a) Exploratory testing is based on user perspective of the software usage
b) Software specification or use-cases are not required to perform exploratory testing
c) Software tester should have a good sense of the usage of software system under test to perform exploratory testing
d) Since exploratory testing is about exploring how the software is written and then design test cases accordingly, code is needed to perform exploratory testing

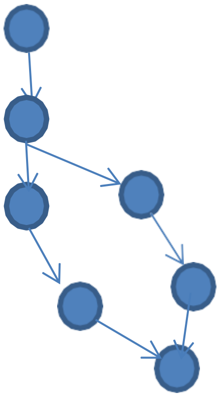
vi.) Suppose you are testing a program that reads a number between 0 and 100 and displays following two messages. If the number is between 1 and 9 display the message "SINGLE DIGIT NUMBER". If the number is between 10 and 99 display the message "DOUBLE DIGIT NUMBER". Any other number should prompt the message "OUT OF RANGE".

Which of the following represents the equivalent partition-based test data?

A) -1, 5, 56, 102 B) 0, 1, 9, 99 C) 1, 9, 10, 99

D) None of the above

vii.) How many independent paths are there in the following control flow graph?



a)2 b) 1 c) 3 d) 4 e) 5

2. Short Answer Questions

Suppose you are testing a software system that keeps records of over-the-counter medications for a pharmacy. The software scans the label of the medication that has the medication name and the volume in ounces. The name of the medication can contain only upper-case letters, lower case letters and a space between words. Spaces at the beginning and the end will be ignored. Also, the valid length range of the medication name is 5 to 20 characters. The volume must be between 1 and 16 ounces. Medications with labels violating these restrictions will be rejected.

a) Develop equivalent partitions for the equivalent partition-based testing for this application (both valid and invalid) Medication name should have minimum five equivalent partitions and the volume should have minimum three equivalent partitions

b) Develop weak normal test cases to test this application.

Test Design Question (More about code based testing will be covered during the lecture on Wednesday 19th and Monday 24th)

3.) **Code Based Testing:** Consider the following modified CountGTG method that counts number of C's in a string that starts with the character 'A'.

```
  public static int CountGTG(String myString)
       {
          int index =0, i=0;
         char A='A', G='G', T='T';
         if(myString.charAt(index++) == A)
         {
           while(index < myString.length()-1)
           {
                if(myString.charAt(index++) == G)
                     {
                         if(myString.charAt(index++)  == T)
                              {
                                      if(myString.charAt(index++)  == G)
                                      {
                                              i++;
                                      }
                              }
                     }

            }

          }
         return i;
}
```

a) Draw the control flow graph for the above function (you can label the above code as needed)

b) Design test cases to perform basic path testing (clearly indicate which test case covers which path)