TUS
Technological University of the Shannon:
Midlands Midwest
Ollscoil Teicneolaíochta na Sionainne:
Lár Tíre Iarthar Láir

TECHNOLOGICAL UNIVERSITY OF THE SHANNON: MIDLANDS MIDWEST

SUMMER EXAMINATION 2022

MODULE: SODV06001 Software Testing

PROGRAMME(S):

LC_KGDVM_KTH	Bachelor of Science (Honours) Games Design and Development
LC_KSFDM_KMY	Bachelor of Science (Honours) Software Development
LC_KSFDM_ITH	Higher Certificate in Science Software Development
LC_KISYM_JMY	Bachelor of Science Internet Systems Development
LC_KISYM_KMY	Bachelor of Science (Honours) Internet Systems Development
LC_KIDMM_KMY	Bachelor of Science (Honours) Interactive Digital Media
LC_KCPTM_JMY	Bachelor of Science Computing

YEAR OF STUDY: 2

EXAMINER(S):

Mr. Brendan Watson (Internal)
Mr. Andrew Shields (External)

TIME ALLOWED: **2 HOURS**

INSTRUCTIONS: **Answer any 3 questions. All questions carry equal marks and marks will be scaled to 100.**

PLEASE DO NOT TURN OVER THIS PAGE UNTIL YOU ARE INSTRUCTED TO DO SO. The use of programmable or text storing calculators is expressly forbidden. Please note that where a candidate answers more than the required number of questions, the examiner will mark all questions attempted and then select the highest scoring ones.

There are no additional requirements for this paper.

Question 1 **(Total 33 Marks)**

a) Explain your understanding of the goal of software testing and the implications of the goal of software testing.

(11 marks)

b) Do you think scenarios and use cases can be used for software testing, explain your answer?

(11 marks)

c) Explain your understanding of the big bang approach to software testing. Do you think you will ever use this approach in the future, explain your answer?

(11 marks)

Question 2 **(Total 33 Marks)**

a) Develop a control flowgraph for the code shown in Figure 1 below and determine the complexity. Suppose software testing has been employed so that TER1 = 1 and TER2 = 1, would you recommend further testing and explain your answer. **(11 marks)**

b) Develop the branch table for the code shown in Figure 1 below.

(11 marks)

c) Develop the block table for the code shown in Figure 1 below.

(11 marks)

```
20      public void walkFirstColOfGridEatingPies(Grid aGrid)
21      {
22          initialise();
23          for (int i=1; i<=2; i++)
24          {
25              turn ("right");
26          }
27
28          for (int j=1; j<=2; j++)
29          {
30              if(aGrid.pieInSight(this)==true)
31              {
32                  eatPie(aGrid);
33              }
34              else
35              {
36                  walk(aGrid);
37              }
38          }
39      }
```

Figure 1

Question 3 **(Total 33 Marks)**

An OvertimeHoursProcessor component has a method called processOvertimeHours which contains business logic about processing of overtime hours worked. The code for the processOvertimeHours is shown in Figure 2 below.

```
package com.mycompany.overtimehoursprocessor;
import java.util.Calendar;

public class OvertimeHoursProcessor
{
    public OvertimeHoursProcessor() { }    // Default constructor

    public Boolean processOvertimeHours(String overtimeHoursFile)
    {
        // First piece of business logic is to check the overtimeHoursFile has
        // a vaild extension.
        if (overtimeHoursFile.endsWith(".data"))
        {
            //Next piece of business logic is to check that it is a Saturday
            // as hours worked this day are overtime rate.
            Calendar cal = Calendar.getInstance();
            if(cal.get(Calendar.DAY_OF_WEEK)==Calendar.SATURDAY))
                {
                    readTheOvertimeHoursFile();
                    return true;
                }
            else
                {
                    return false;
                }
        }
        else
        {
           return false;
        }
    }

        public void readTheOvertimeHoursFile()
        {
        // This code is under construction and is not currently needed
        // to unit test the business logic in the processOvertimeHours method.
        }
}
```

Figure 2

a) Explain what a stub is and why you need to utilize stubs to unit test code. **(8 Marks)**

b) Refactor the OvertimeHoursProcessor to make it testable by introducing a layer of indirection to avoid the dependency i.e. write code or pseudocode. You refactoring should include adding an interface which will allow use of a configurable stub in the unit tests. **(12 Marks)**

c) Write code or pseudocode for three unit tests to test the business logic in the processOvertimeHours method. Write code or pseudocode for a configurable stub to be used by your tests utilising constructor injection. **(13 Marks)**

Question 4 **Total 33 Marks**

a) Explain your understanding of equivalence partitioning.

(10 Marks)

b) A software system to accept new stock items in a steel yard accepts the item name followed by a list of different lengths the steel comes in. The specification states that the item name is to be alphabetic 5 to 10 characters long. Each length in metres must be in the range of 1 to 7, whole numbers only. The lengths are to be entered in descending order (biggest length first) with a maximum of 3 lengths allowed to be entered for each item and whole numbers only. A comma is to be used to separate the item name from the lengths and a comma will be used to separate each length and the enter key to be pressed after the last length is entered
Derive the equivalence classes and determine black box test cases based on these and utilise boundary value analysis. **(23 Marks)**