LIMERICK INSTITUTE OF TECHNOLOGY
INSTITIÚID TEICNEOLAÍOCHTA LUIMNIGH

LIMERICK INSTITUTE OF TECHNOLOGY

OPEN BOOK SUMMER EXAMINATION 2021

MODULE: SODV06002-Software Testing

PROGRAMME(S):

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

YEAR OF STUDY: 2

EXAMINER(S):

Brendan Watson	(Internal)
Tom Davis	(Internal)
Mr. Andrew Shields	(External)
Dr. Bianca Schoen-Phelan	(External)
Dr. Markus Hofmann	(External)

TIME ALLOWED: **3 HOURS**

INSTRUCTIONS: **Answer any 3 questions. All questions carry equal marks and marks will be scaled to 100.**
Upload a Word file containing your answers to the questions.

QUESTION 1 ***[TOTAL MARKS: 33]***

Question 1(a) **[11 Marks]**

Explain in your own words both software testing and software debugging using the following four headings: Objective, Output, Knowledge of Design, Responsibility (who's job).

Question 1(b) **[11 Marks]**

Many of the unit testing practical/computer lab sessions in this module included both the static and dynamic approaches to software testing. Do you think this was useful, explain your answer in your own words.

Question 1(c) **[11 Marks]**

Explain in your own words your understanding of Regression testing. What kind of tests would you expect to find in a regression test suite?

QUESTION 2 ***[TOTAL MARKS: 33]***

Question 2(a) **[7 Marks]**

Explain the following Test Effectives Ratios: TER1, TER2. Explain in your own words how determining the value for the complexity metric for a method can help you to achieve complete coverage.

Question 2(b) **[10 Marks]**

Develop the branch table for the code shown in Figure 1 below. Suppose software testing has been employed so that TER1 = 1 and TER2 = 0.8, would you recommend further testing? Explain your answer in your own words.

Question 2(c) **[6 Marks]**

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

Question 2(d) **[10 Marks]**

Explain in your own words why you think it is useful to develop a block table.

```
8           Grid theGrid = new Grid(12,12,"MyGrid");
9           Pie aPie = new Pie(1,9,"Pie_1");
10          PieEater aPieEater = new PieEater(3,3,"PieEater_A", "East");
11          aPieEater.initialise();
12          for(int i=1; i<=12; i++)
13          {
14              if(theGrid.pieInSight(aPieEater)==true)
15              {
16                  aPieEater.eatPie(theGrid);
17              }
18              else
19              {
20                  aPieEater.walk(theGrid);
21              }
22          }
23          if (aPieEater.getPieCount()==1)
24          {
25              aPieEater.spinAndEatPie(theGrid);
26          }
27          else
28          {
29              aPieEater.initialise();
30          }
31        aPieEater.reportStatus();
```

Figure 1

QUESTION 3 ***[TOTAL MARKS: 33]***

A MonthlyStockProcessor component has a method called processStock which contains business logic about processing of monthly stock. The code for the processStock is shown in Figure 2 below.

```
package StockSystem;
import java.util.Calendar;

public class MonthlyStockProcessor
{
    public MonthlyStockProcessor() { }// default constructor
    public Boolean processStock(String dataFile)
        {
         // First piece of business logic is to check that the dataFile has a valid extension.
            if(!dataFile.endsWith(".dat")) {
                return false;  //dataFile extension is invalid
            }
            // Next piece of business logic is to check that it's last day of
            // the month
            Calendar cal = Calendar.getInstance();
            int maxDay = cal.getActualMaximum(Calendar.DAY_OF_MONTH);
            // maxDay stores the number of days in the current month.
            if(cal.get(Calendar.DAY_OF_MONTH) == maxDay) {
                readTheDataFileAndProcessStock();
                return true;
            }
            else {
                return false;
            }
        }

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

Figure 2

Question 3(a) **[9 Marks]**

Refactor the MonthlyStockProcessor 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.

Question 3(b) **[9 Marks]**

Write code or pseudocode for three unit tests to test the business logic in the processStock method. Write code or pseudocode for a configurable stub to be used by your tests utilising constructor injection.

Question 3(c) **[15 Marks]**

Explain in your own words how you achieved unit test of the code in Figure 2.

QUESTION 4 ***[TOTAL MARKS: 33]***

Question 4(a) **(3 marks)**

What is the purpose of Black box testing?

Question 4(b) **(8 marks)**

Explain with the use of a simple example why it is necessary to use both functional and structural testing methods when developing software.

Question 4(c) **(8 marks)**

Explain the relationship between Equivalence Classes/Partitions and Boundary Value Analysis. Use a simple example to illustrate your answer.

Question 4(d)

A shopping website offers different discounts depending upon each transaction made by the customer. For example if a purchase is in the range €1 to €50 then there is no discount, a purchase over €50 and up to €200 has 5% discount. Purchases between €201 and up to €500 have a 10% discount, and purchases over €501 have a 15% discount.

(i) Identify all relevant equivalence classes. **(4 marks)**

(ii) Using Boundary Value Analysis identify the boundary values for each Equivalence Class. **(4 marks)**

(iii) List the Black Box Test Cases using the Equivalence Classes and Boundary Values. **(6 marks)**