



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 2023

MODULE: SODV06010 Software Testing

PROGRAMME(S):

LC_KMWCM_KMY	Bachelor of Science (Honours) Mobile and Web Computing
LC_KISYM_KMY	Bachelor of Science (Honours) Internet Systems Development
LC_KCPTM_JMY	Bachelor of Science Computing
LC_KSOFM_KMY	Bachelor of Science (Honours) Software Development
LC_KISYM_JMY	Bachelor of Science Internet Systems Development
LC_KMWCM_JMY	Bachelor of Science Mobile and Web 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) Do you think that all bugs can be removed by software testing? Explain your answer and give your understanding of the goal of software testing in your explanation.

(11 marks)

b) Do you think scenarios and use cases can be used for black box testing? Explain your answer?

(11 marks)

c) Explain your understanding of integration testing. Do you think you will use integration testing 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)

```
91      for(int i=1; i<=2; i++)
92      {
93          for(int j=1; j<=2; j++)
94          {
95              ThirdPieEater.turn("Right");
96              if (aGrid.pieInSight(ThirdPieEater)==true)
97              {
98                  ThirdPieEater.eatPie(aGrid);
99              }
100             else
101             {
102                 ThirdPieEater.reportStatus();
103             }
104         }
105         if (aGrid.clearAhead(ThirdPieEater)==true)
106         {
107             for(int j=1; j<=2; j++)
108             {
109                 ThirdPieEater.walk(aGrid);
110             }
111         }
112     }
113     ThirdPieEater.reportStatus();
```

Figure 1

Question 3

(Total 33 Marks)

An InvoiceLoader component has a method called loadInvoices which contains business logic about processing of invoice data. The code for the loadInvoices is shown in Figure 2 below.

```
1 package com.mycompany.invoiceLoader;
2 import java.util.Calendar;
3 public class InvoiceLoader
4 {
5     public InvoiceLoader() { } // default constructor
6
7     public Boolean loadInvoices(String dataFile)
8     {
9         // First piece of business logic is to check that the dataFile has a valid extension.
10        if(!dataFile.endsWith(".data"))
11        {
12            return false; //dataFile extension is invalid
13        }
14        // Next piece of business logic is to check that it's Friday as this is the only day of the
15        // week that invoices are loaded
16        Calendar cal = Calendar.getInstance();
17        if(cal.get(Calendar.DAY_OF_MONTH) == Calendar.FRIDAY)
18        {
19            readTheDataFileAndLoadTheInvoices();
20            return true;
21        }
22        else
23        {
24            return false;
25        }
26    }
27
28    public void readTheDataFileAndLoadTheInvoices ()
29    {
30        // This code is under construction and is not currently needed
31        // to unit test the business logic in the loadInvoices method.
32    }
33 }
34
```

Figure 2

a) When you develop production code you will need seams to ensure your code is testable, explain what a seam is and how you will use them in your unit tests.

(8 Marks)

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

(10 Marks)

b) A shopping website offers different discounts depending upon each transaction made by the customer. The user enters the purchase amount, then presses the OK button. For example if a purchase is in the range €1 to €50 then no discount is displayed, a purchase over €50 and up to €200 have 5% discount calculated and displayed. Purchases between €201 and up to €500 have 10% discount calculated and displayed, and purchases over €501 have a 15% discount calculated and displayed. The maximum purchase amount that can be entered is €1000 and the minimum amount is €1 and invalid input of any kind results in a warning message Invalid Input.

Derive the equivalence classes and determine black box test cases based on these and utilise boundary value analysis.

(23 Marks)