LIMERICK INSTITUTE OF TECHNOLOGY
INSTITIÚID TEICNEOLAÍOCHTA LUIMNIGH

LIMERICK INSTITUTE OF TECHNOLOGY

SUMMER EXAMINATION 2019

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):

Desmond O Carroll	(Internal)
William Ward	(Internal)
Brendan Watson	(Internal)
Tom Davis	(Internal)
Mr. Andrew Shields	(External)
Dr. Bianca Schoen-Phelan	(External)
Dr. Markus Hofmann	(External)

TIME ALLOWED: **2 HOURS**

INSTRUCTIONS: **Answer 3 questions. All questions carry equal marks.**

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.

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

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

The goal of software testing refers to an acceptable level of confidence and to correct system behaviour, explain what both acceptable level of confidence and correct system behaviour mean in this context.

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

Explain both the static and dynamic approaches to software testing.

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

Explain system testing. What is the big bang approach to software testing and why do you think some developers actually use this approach.

QUESTION 2 *[TOTAL MARKS: 33]*

Question 2(a) [7 Marks]

Do you agree that professional software developers should test their software so that they achieve complete coverage? Explain your answer.

Question 2(b) [10 Marks]

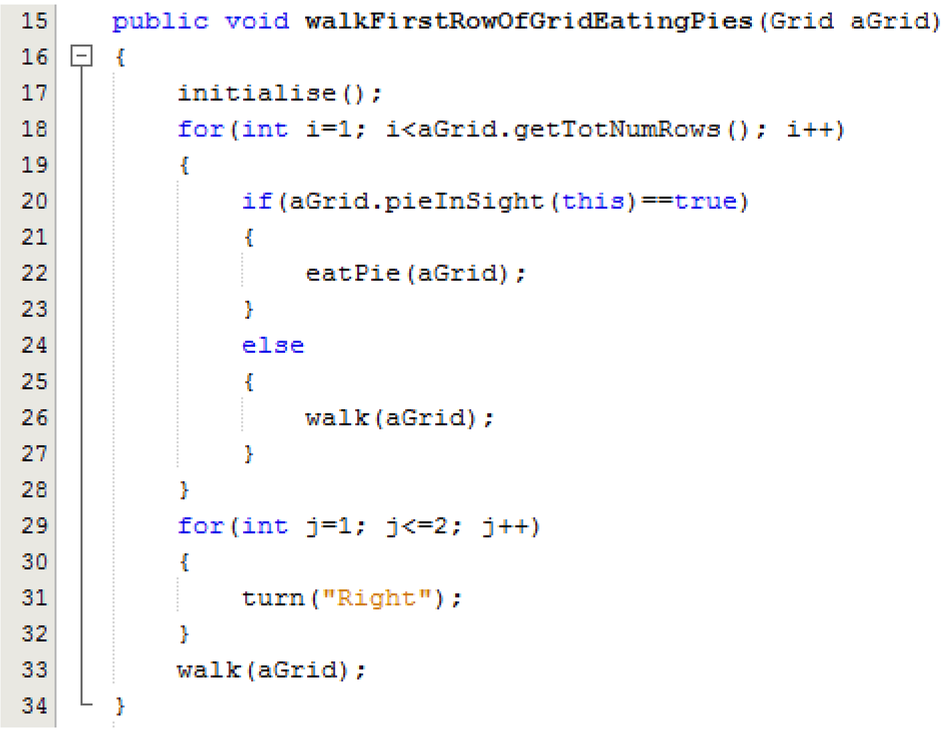
Develop a control flowgraph for the code shown in Figure 1 below and determine the complexity. Suppose software testing has been employed so that TER1 = 1 and TER2 = 0.6, would you recommend further testing? Explain your answer.

Question 2(c) [8 Marks]

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

Question 2(d) [8 Marks]

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

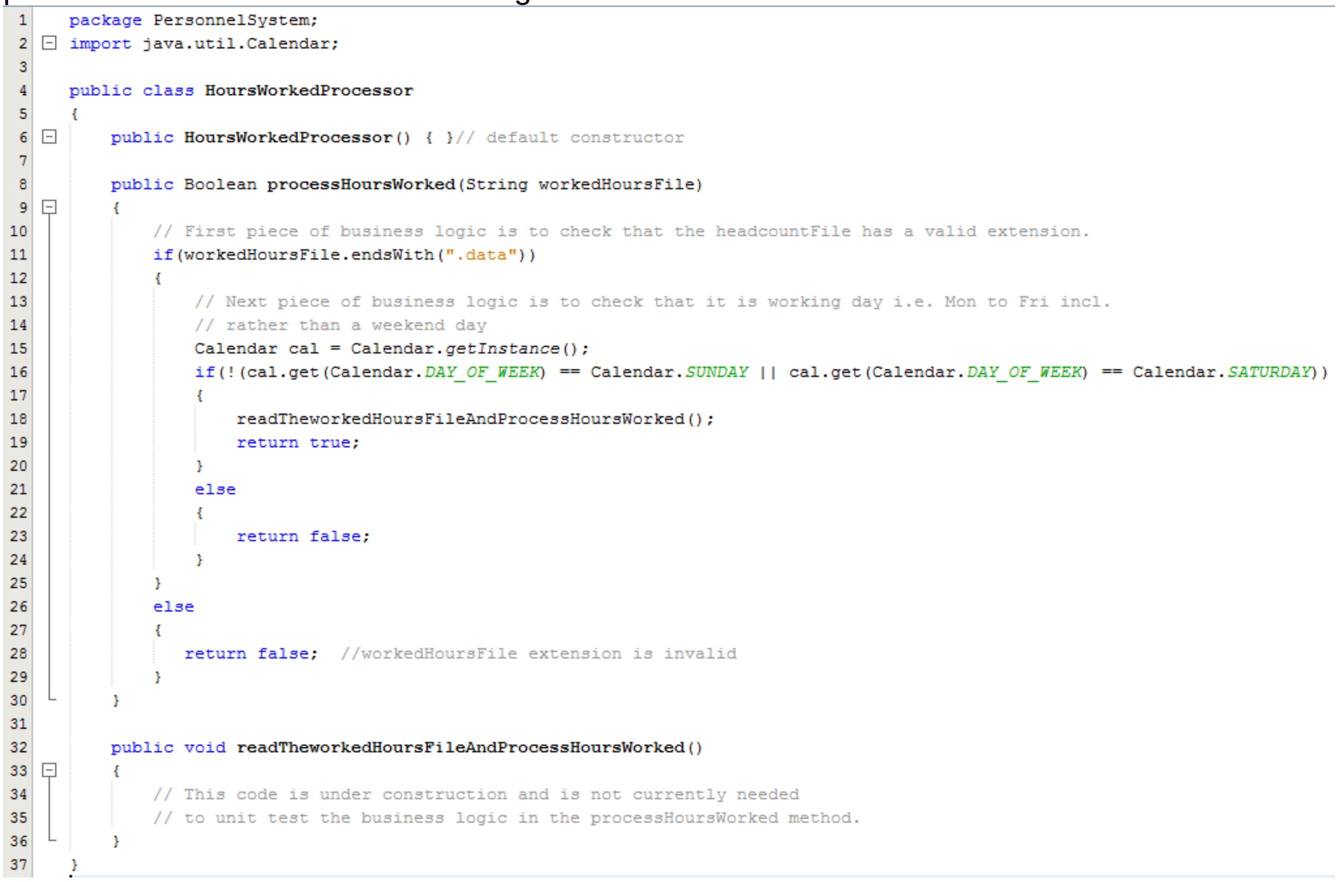


```
public void walkFirstRowOfGridEatingPies(Grid aGrid)
{
    initialise();
    for(int i=1; i<aGrid.getTotNumRows(); i++)
    {
        if(aGrid.pieInSight(this)==true)
        {
            eatPie(aGrid);
        }
        else
        {
            walk(aGrid);
        }
    }
    for(int j=1; j<=2; j++)
    {
        turn("Right");
    }
    walk(aGrid);
}
```

Figure 1

QUESTION 3 *[TOTAL MARKS: 33]*

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



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

public class HoursWorkedProcessor
{
    public HoursWorkedProcessor() { }// default constructor

    public Boolean processHoursWorked(String workedHoursFile)
    {
        // First piece of business logic is to check that the headcountFile has a valid extension.
        if(workedHoursFile.endsWith(".data"))
        {
            // Next piece of business logic is to check that it is working day i.e. Mon to Fri incl.
            // rather than a weekend day
            Calendar cal = Calendar.getInstance();
            if(!(cal.get(Calendar.DAY_OF_WEEK) == Calendar.SUNDAY || cal.get(Calendar.DAY_OF_WEEK) == Calendar.SATURDAY))
            {
                readTheworkedHoursFileAndProcessHoursWorked();
                return true;
            }
            else
            {
                return false;
            }
        }
        else
        {
           return false;  //workedHoursFile extension is invalid
        }
    }

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

Figure 2

Question 3(a) [8 Marks]

Explain what a stub is. What are seams in code?

Question 3(b) [12 Marks]

Refactor the HoursWorkedProcessor 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(c) [13 Marks]

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

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

Question 4(a) **[4 Marks]**

What is the purpose of Black box testing?

Question 4(b) **[10 Marks]**

Briefly explain the following terms and give an example for each.

(i) Equivalence Partitioning.

(ii) Boundary Value Analysis.

Question 4(c) **[19 Marks]**

Consider a software module that is intended to accept the name of a grocery item and a list of the different sizes the item comes in, specified in ounces. The specifications state that the item name is to be alphabetic characters 2 to 15 characters in length. Each size may be a value in the range of 1 to 48, whole numbers only. The sizes are to be entered in ascending order (smaller sizes first). A maximum of five sizes may be entered for each item. The item name is to be entered first, followed by a comma, then followed by a list of sizes. A comma will be used to separate each size. Spaces (blanks) are to be ignored anywhere in the input.

For the module described above:

(i) Derive the Equivalence Classes.

(ii) List the Black Box Test Cases based on the Equivalence Classes.