Programme Code: TU856, TU858
Module Code: CMPU2016
CRN: 23094, 22399

TECHNOLOGICAL UNIVERSITY DUBLIN

CITY CAMPUS - GRANGEGORMAN

TU856 – BSc. (Honours) in Computer Science
TU858 – BSc. (Honours) in Computer Science (International)

Year 2

SEMESTER 2
EXAMINATIONS 2023/24

Object Oriented Programming

Internal Examiner:
Dr. Lucas Rizzo
Dr. Bryan Duggan
Dr. Paul Doyle

External Examiner:
Ms. Pamela O'Brien

Exam Duration: 3 hours

Instructions:
There are 2 sections on the paper: section A and section B. Candidates must answer two questions out of each section. Answer four questions in total. All questions carry equal marks.

SECTION A
OO Programming through Python

Answer any TWO questions of the THREE Questions in this Section

1. (a) Analyse the Python class below:

```
class Point:
    def __init__(self, x, y):
        self.x = x
        self.y = y

    def __add__(self, other):
        return Point(self.x + other.x, self.y + other.y)

    def __str__(self):
        return "Point({}, {})".format(self.x, self.y)
```

(i) What is the output of the following code and what is the data type of p3? **(2 marks)**

```
p1 = Point(1, 2)
p2 = Point(3, 4)
p3 = p1 + p2
print(p3)
```

(ii) Why does the following code crash? **(3 marks)**

```
p4 = Point(5, 7)
p5 = p4 + (6, 9)
print(p5)
```

(iii) Revise the `__add__` method to enhance its functionality. Incorporate introspection techniques to enable the method to process 2-element tuples, interpreting the first element as the x-coordinate and the second as the y-coordinate, effectively transforming it into a Point object. The method should reject tuples of any length other than 2 by raising an `AttributeError`. **(10 marks)**

(b) Analyse the Python code below:

```
def char_frequency(str1):
    """Function that counts the number of
characters in a string"""
    dict = {}

    for n in str1:
        if n in dict.values():
            dict[n] += 1
        else:
            dict[n] = 1
    return dict
```

(i) The function above does not count the number of characters in a string as expected. Modify it so it returns the correct output. **(2 marks)**

(ii) Modify the function again so that it only counts characters from a to z. The function should not be case sensitive. Assume the following string is given: **(8 marks)**

```
import string

letters_a_to_z = string.ascii_lowercase
```

2. (a) Analyse the code below:

```
str_list = ['hi','mom','dad']
num_list = [1,57,15]
num_list[-1] = 25

print(str_list + num_list) # Line 1
print([str_list[0],num_list[-1]]) # Line 2
print(str_list.append(num_list)) # Line 3
print(str_list) # Line 4
print(str_list.sort())# Line 5
```

(i) What is the output of Line 1? **(2 marks)**
(ii) What is the output of Line 2? **(2 marks)**
(iii) What is the output of Line 3? **(2 marks)**
(iv) What is the output of Line 4? **(2 marks)**
(v) What is the output of Line 5? **(2 marks)**

(b) (i) Design a class focused on managing Olympic medal awards. This class should track the following information for each medal: the athlete's name, their country, the event, and the type of medal (gold, silver, or bronze). Include a `__str__` method to enable easy printing of the medal details. **(5 marks)**

(ii) Overload the ">" operator by implementing the `__gt__` method in your class. A gold medal should be greater than a silver medal and a bronze medal. A silver medal should be greater than a bronze medal. A bronze medal is always the smallest. **(5 marks)**

(iii) Implement a new SpecialAchievementMedal class that inherits from Medal. The special achievement medal should have the same attributes as Medal plus a special reason (implemented as a string) as one of its attributes. Implement the class with a constructor and `__str__` methods. **(5 marks)**

3. **(a)** Analyse the Python code below:

```
my_dict = {'bill': 110, 'zach': 'hi mom',
'laurie': 'bye mom'}

try:
    result = ''
    key_str = input("Enter a key:")
    val = my_dict[key_str]
    result = result + val
    result = int(result) + 10
except KeyError:
    result = 'hi mom'
except TypeError:
    result = '100'

print(result)  # Line 1
```

(i) What output does Line 1 produce with the input 'bill' and why? **(3 marks)**

(ii) What output does Line 1 produce with the input '110' and why? **(3 marks)**

(iii) For the input `'zach'` the code will raise a `ValueError`, which has not been implemented. Explain why this error happens and implement a `ValueError` exception. The output for Line 1 should be `'hi zach'` when the input is `'zach'`. In order words, the output should be the string `'hi '` concatenated to the user input. **(4 marks)**

(b) Suppose you are a space colony administrator tasked with building and managing a thriving colony of ants on a distant planet or asteroid.

(i) Define a Python class named `Equipment` for representing tools that can be used by ants in the colony. **(6 marks)**

Include attributes for name (string), function (string), and durability (int). Implement methods to check durability, perform maintenance (by increasing durability), and use the equipment (by reducing durability). You can define any values for setting and changing durability but ensure that an equipment cannot be used if durability reaches 0 or less.

(ii) Design a Python class named `WorkerAnt` to model individual worker ants in the colony. **(6 marks)**

Include attributes for name (string), age (integer), and a list of owned equipment. The `WorkerAnt` and `Equipment` have an aggregation relationship. Implement methods to add equipment and perform tasks using some equipment by name.

(iii) In the main scope, create one instance of `WorkerAnt` and one instance of `Equipment`. Add the equipment to the worker and call the method implemented to simulate its use. **(3 mark)**

SECTION B
OO Programming through Java

Answer any TWO questions of the THREE Questions in this Section

Question 4

(a) Figure 1 shows a screenshot from YASC, a game inspired by Spacewar:

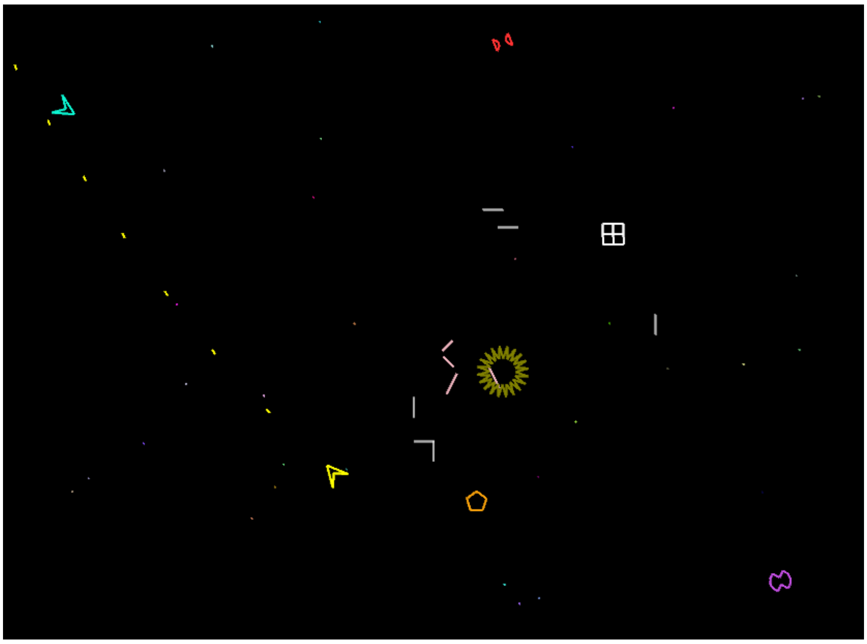

Figure 1

Give examples of how the following OO programming principles are employed in the development of this game.

(i) Encapsulation
(ii) Inheritance
(iii) Polymorphism
(iv) Interfaces
(v) Abstract classes

(5 x 3 marks)

(b) Draw a class diagram showing how the classes in game relate to each another. Include examples of *inheritance* and *aggregation* in your solution.

(5 marks)

(c) Figure 2 shows code that iterates over an ArrayList of GameObjects:

```
for(GameObject go:gameObjects)
{
    go.update();
    go.render();
}
```

Figure 2

The update method on GameObject might remove objects from the ArrayList. What is wrong with this code and how would you fix it?

(5 marks)

Question 5

(a) The Game of Life, is a *cellular automata* devised by the British mathematician John Horton Conway in 1970. What is a *cellular automata*? Where else are they used in computer science?

(5 marks)

(b) Write a class called Board in Java to encapsulate the Game of Life board. Include in your solution:

(i) Private fields.
(ii) A public constructor that takes two parameters: the size of the board and a reference to a PApplet.
(iii) A public accessor method to return the value held in a cell. This method should incorporate bounds checking.
(iv) A private method to count the alive cells around a cell.
(v) A public method to draw the board using the Processing libraries.

(5 x 4 marks)

Question 6

(a) Explain the following terms: *sample rate*, *resolution*, *frame size*, *spectrum*, *channels*.

(5 marks)

(b) What does *forking* a git repository mean? How do you set the upstream on a forked repository?

(4 marks)

(c) What is a *merge conflict*? How would you resolve a merge conflict on:

(i) A CSV file
(ii) An MP3 file

(6 marks)

(d) Explain *lerp* and *map* from the Processing library. Give an example of the usage of these functions.

(6 marks)

(e) What is an *interface* in Java? Give an example.

(4 marks)