

## National College of Ireland

Higher Diploma in Science in Computing (HDSDEV\_JANOLY1\_O)  
Higher Diploma in Science in Computing (HDSDEV\_JAN21OL)  
Higher Diploma in Science in Computing (HDSDEV\_INT)  
Higher Diploma in Science in Computing (HDSDEV\_SEP)  
Higher Diploma in Science in Computing (HDSDEV\_SEPOL\_YR1)  
Higher Diploma in Science in Computing (HDCSDEV\_INTJAN22)  
Higher Diploma in Science in Computing (HDSDEV\_JAN22)

### Semester Three Examinations 2021/22

Release Date: Saturday the 20<sup>th</sup> of August 2022, 10:00  
Submission Date: Saturday the 20<sup>th</sup> of August 2022, 20:00

---

### Data Structures

Jack O'Neill

Keith Maycock, Emin Zerman, Tobi Ekundayo, John Andrews

EugeneMcLaughlin

Enda Stafford

Attempt **all** Questions

Duration of Exam: 10 hours

Attachments: None

**Instructions:** Questions 1 and 2 should be provided as a Word document. Questions 3 and 4 should be completed using Java. You need to provide extensive commenting to demonstrate your clear understanding of your solutions. You should upload a single compressed folder (.zip) consisting of all your files to Moodle prior to 8 pm on Saturday 20<sup>th</sup> August 2022.

This assessment is an open book assessment and **NOT a collaborative exercise**. All submission will be subjected to comparison for plagiarism to ensure the integrity of the assessment. It should also be noted that an examiner may require to conduct a viva examination with you to further clarify any concerns with any submission.

## PART 1 - THEORY

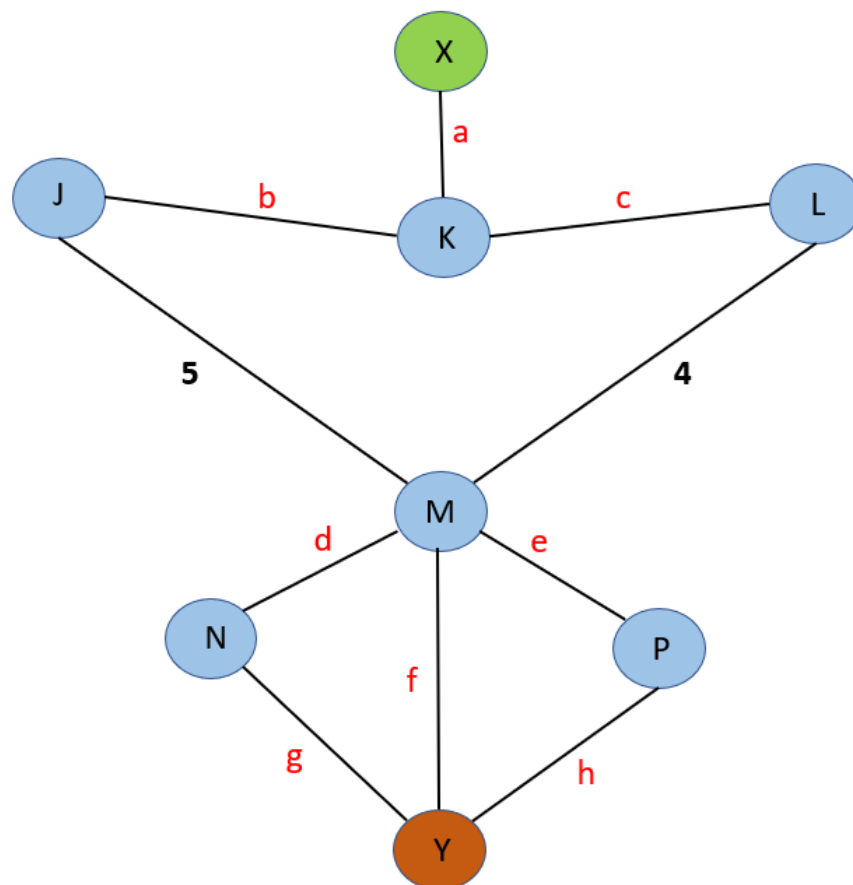
### Question 1

Using Dijkstra's shortest path algorithm calculate the shortest path from **Node X** through to **Node Y** in the diagram below. You should clearly show each step of the algorithm.

Find the appropriate weights using your student ID in the following order: abcdefgh. (e.g., if your student ID is 12345678, then a=1, b=2, c=3, ..., g=7, h=8). Afterwards, calculate the shortest path from **Node X** to **Node Y** using Dijkstra's shortest path algorithm.

Please provide a Word document for the solution of this question. (Hint: You can draw the solution on a physical page using a pen. You can take a photo of this paper and attach it in the Word document.)

[15 marks]



### Question 2

Investigate the evolution of Generics in Java, since their introduction in Java.-5. Give examples of the benefits of their usage, using exemplification. This is a theoretical analysis, no coding is required for this question. Guideline for your answer is 500 – 800 words.

[10 marks]

## PART 2 - PRACTICAL

### Question 3

"XML text editors are great at helping you write xml files. They can do syntax highlighting and make sure that you have a closing tag, `</>`, for each opening tag `<>`."

Develop and implement a data structure as described below for the implementation of a validator that is capable of reading in an xml file to check if each open tag has a corresponding closing tag.

**[30 marks total]**

This question includes assessment variability, and you should attempt the questions considering the last digit of your own student ID. If you answer the wrong section, you will be awarded 0 marks.

**You should provide functionality for the following artefacts:**

If the **last digit** of your student ID is **ODD (1, 3, 5, 7, 9)**, provide an implementation using a **Queue** data structure.

1. An interface that defines the following methods:

- enqueue
- dequeue
- peek
- toString

**[3 marks]**

2. A node class that can be used as the basis of your data structure.

**[3 marks]**

3. Implement a data structure that will hold the occurrences of opening and closing tags, `<>` and `</>` and provides an implementation for the actions defined in the above interface.

**[16 marks]**

4. Prepare a code piece to test your implementation.

**[8 marks]**

If the **last digit** of your student ID is **EVEN (0, 2, 4, 6, 8)**, provide an implementation using a **Stack** data structure.

1. An interface that defines the following methods:

- push
- pop
- peek
- toString

**[3 marks]**

2. A node class that can be used as the basis of your data structure.

**[3 marks]**

3. Implement a data structure that will hold the occurrences of opening and closing tags, `<>` and `</>` and provides an implementation for the actions defined in the above interface.

**[16 marks]**

4. Prepare a code piece to test your implementation.

**[8 marks]**

#### Question 4

As a Java developer, you are tasked to develop data structures for a newspaper article on Airport delays. The newspaper article needs to sort the airports based on an Airport Waiting Index. Since you have recently been working on trees, you decide to create trees for this problem, in order to sort the input data. For this, there are several steps you need to take.

[45 marks total]

Define a suitable comparable **Airport** object that contains the following information: Airport Name, Airport Location and an Airport Waiting Index (integer assigned between the range of 1 - 10, with short-wait airport/items closer to 1, (best Airport Waiting Index), and long-wait airports/items closer to 10, worst Airport Waiting Index). Comparison of Airport Objects should use the Airport Waiting Index attribute.

[5 marks]

- a) Interfaces allow us to make sure a number of methods need to be implemented when implementing the Interface. Write a suitable interface for a Binary Tree that will ensure that only trees of comparable objects can be constructed. You should provide methods to:
1. Check if the tree is empty
  2. To add Nodes into the tree
  3. To count all the Nodes
  4. Find the best **Airport** object (least wait time/ lowest Airport Waiting Index)
  5. Find the worst **Airport** object (worst wait time/ highest Airport Waiting Index)

[5 marks]

- b) Define a suitable Node for a Binary Tree using Java Generics.

[5 marks]

- c) Provide an implementation of the Interface defined in part a) (Binary Tree implementation).

[5 \* 5 marks]

- d) Create and add ten **Airport** objects to clearly demonstrate the functionality of the Binary Tree.

[5 marks]