

ASSIGNMENT 1 FRONT SHEET

Qualification	BTEC Level 5 HND Diploma in Computing		
Unit number and title	Unit 19: Data Structures and Algorithms		
Submission date		Date Received 1st submission	
Re-submission Date		Date Received 2nd submission	
Student Name		Student ID	
Class		Assessor name	Hong-Quan Do
Student declaration <p>I certify that the assignment submission is entirely my own work and I fully understand the consequences of plagiarism. I understand that making a false declaration is a form of malpractice.</p>			
		Student's signature	

Grading grid

P1	P2	P3	M1	M2	M3	D1	D2

☐ **Summative Feedback:**

☐ **Resubmission Feedback:**

Grade:

Assessor Signature:

Date:

Internal Verifier's Comments:

IV Signature:

TRANG BÌA

MỤC LỤC

(GIỚI THIỆU MÔN HỌC)

PART 1

- **P1** Create a design specification for data structures explaining the valid operations that can be carried out on the structures.

The specification consists of the names of every function, the type of its arguments, and the type of its result. There should also be a description of what the function does, but without appealing to internal representation or implementation details.

Example: <https://www.geeksforgeeks.org/abstract-data-types/>

- **P2** Determine the operations of a memory stack and how it is used to implement function calls in a computer.
- **M1** Illustrate, with an example, a concrete data structure for a First In First out (FIFO) queue.

It is necessary to support your evidence with examples and a diagrammatic representation (figures to explain each Queue's operation)

- **M2** Compare the performance of two sorting algorithms.
 - **D1** Analyze the operation, using illustrations, of two network shortest path algorithms, providing an example of each.
-

Here a sample for PART 1

1. ADT (**P1**)

Discussion on ADTs and the value of formal specification of ADTs.

E.g. An abstract data type (ADT) consists of the following:

- ☒ A collection of data
- ☒ A set of operations on the data or subsets of the data
- ☒ A set of axioms, or rules of behavior governing the interaction of operations

2. Stack ADT (P1)

- What is stack? E.g. Linear structure, LIFO
- Stack operations and working mechanism

(Reference:

<https://runestone.academy/runestone/books/published/pythonds3/index.html>)

E.g. Primitive operations of a stack are:

- (a) Insert an element on top of the stack (push),
- (b) Remove the top element from the stack(pop),
- (c) **isEmpty**, determines if the stack is empty.
- (d) **top**, which returns the top item

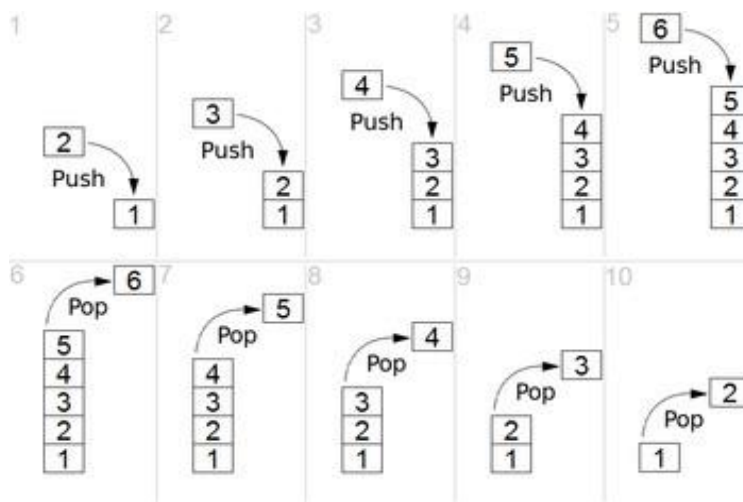


Figure 1. Stack operations: Push() and Pop() (Source: Wikipedia.com, 2020)

- Applications/Examples of Stack

3. Memory Stack (P2)

Determine the operations of a memory stack and how it is used to implement function calls in a computer.

- o The definition and main operations of memory stack

E.g. <https://www.guru99.com/stack-vs-heap.html>

- o How it is used to implement function calls in a computer

Expect a particular example to illustrate your answer.

4. Queue ADT (P1 + M1)

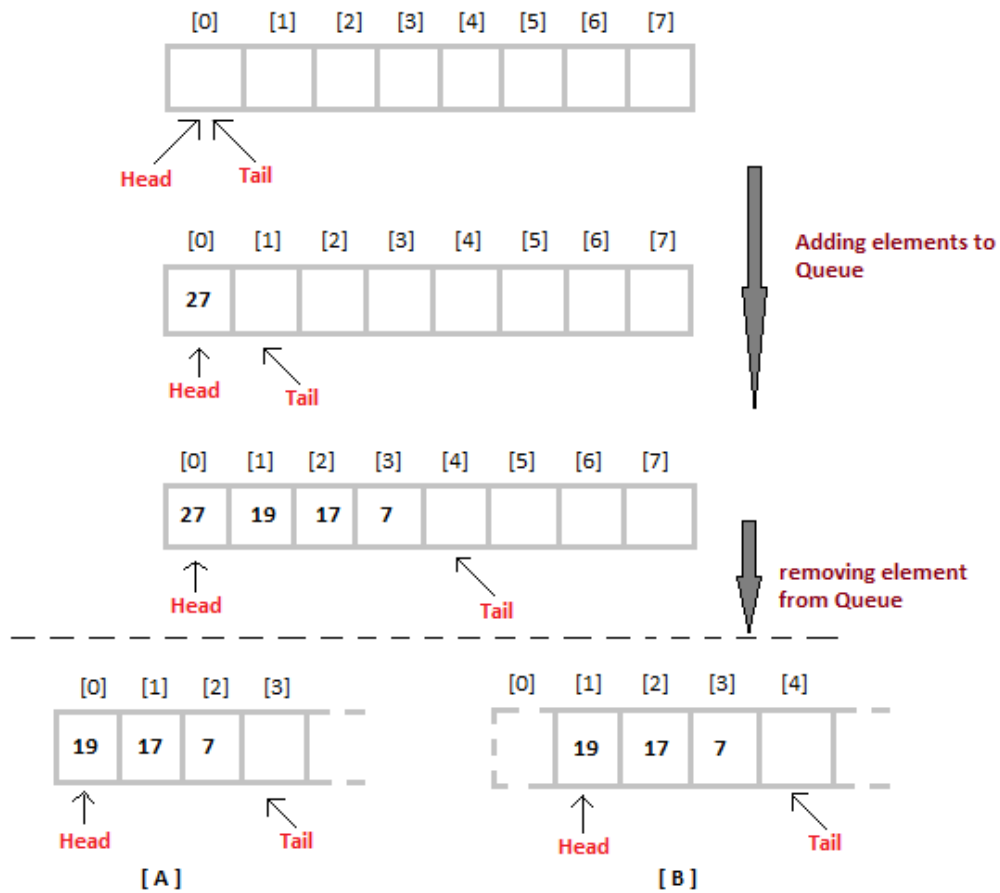


Figure 2. Queue operations: *Enqueue()* and *Dequeue()* (Source: studytonight.com, 2020)

5. Sorting Algorithms (M2)

- o Discuss the purpose of sorting
- o Compare different sort algorithms, explaining their limits and performance.

Ví dụ:

a. Introduction

- Bài toán sắp xếp là gì
- Có những thuật toán điển hình nào: Trong phần b và c, tôi/chúng tôi sẽ trình bày thuật toán:
- Tại sao lại chọn 2 thuật toán này: So sánh 2 thuật toán (Performance, Ưu nhược điểm)

b. Selection Sort

- Trình bày thuật toán: Các bước thực hiện / Mã giả/ Flow chart
- Ví dụ (Tự đưa ví dụ)
- (Optional) Implementation
 - o Data structure used
 - o Time complexity /Space complexity
 - o Source code

c. Quick Sort

6. Two network shortest path algorithms (D1)

- a. Mô tả bài toán
- b. Trình bày thuật toán: Các bước thực hiện / Mã giả/ Flow chart c. Ví dụ.
- d. (Optional) Implementation
 - o Data structure used
 - o Time complexity /Space complexity
 - o Source code

PART 2

- **P3** Using an imperative definition, specify the abstract data type for a software stack.
You should include the elements like ADT specification using formal notation e.g. ASN.1, use of non-executable program specification language e.g. SDL and VDM, Issues e.g. complexity in software development, design patterns, parallelism, interfaces, encapsulation, information hiding, efficiency etc. and creational conditions like pre- conditions, post-conditions and error conditions.
 - **M3** Examine the advantages of encapsulation and information hiding when using an ADT.
 - **D2** Discuss the view that imperative ADTs are a basis for object orientation and, with justification, state whether you agree.
-

Here a sample for PART 2

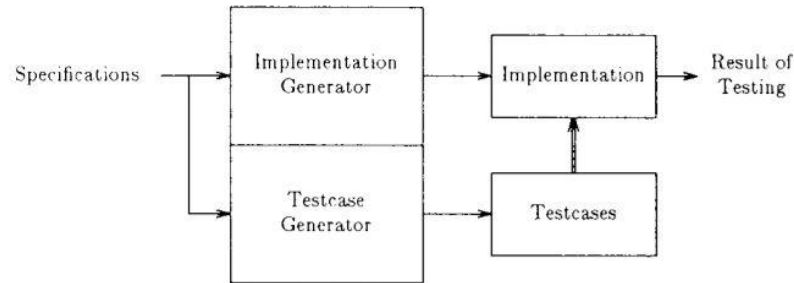
1. Explanation on how to specify an abstract data type using the example of software stack (P3)

Material

1. Slide: <https://web.mit.edu/16.35/www/lecturenotes/FormalMethods.pdf>
2. Book chapter: https://ifs.host.cs.st-andrews.ac.uk/Books/SE9/WebChapters/PDF/Ch_27_Formal_spec.pdf
3. Paper:
 - <https://pdfs.semanticscholar.org/994e/ed27c418113b2ec27bceb25fe70cc3aadcd.pdf>
 - <http://elib.mi.sanu.ac.rs/files/journals/yjor/13/yujorn13p79-96.pdf>
4. An example: <https://learn1.open.ac.uk/mod/oublog/viewpost.php?post=162582>

Sample

- a. Introduction to formal specification, types of formal specification languages (E.g., Axiomatic specifications, VDM).



- b. Describe what are pre-condition, post-condition, error-condition.
- c. Specify Stack's operations using this formal specification language.

Pay attention to write the specification based on your own understanding. The specification should include descriptions of pre-condition, post-condition, error-condition for each operation.

2. Explanation of the advantages of encapsulation and information hiding when using an ADT (M3)

What is encapsulation

Why we need encapsulation?

Advantages and information hiding when using an ADT Encapsulation uses accessors and mutators Encapsulation using properties

Read only property

Write only property

3. Discussion of imperative ADTs with regard to object orientation (D2)

- Object oriented program: 4 concepts – Encapsulation, Abstraction, Polymorphism, and Inheritance. ADTs: Encapsulation, Abstraction
- William R. Cook [1] states that Objects are centered around the constructors of a data abstraction, while abstract data types are organized around the operations. Objects are centered around the constructors of a data abstraction, while abstract data types are organized around the operations, according to William R. Cook [1].
- ADTs vs. object orientation

EVALUATION (D2)

- o Tóm tắt 2 part làm gì.

- o Tự đánh giá: Understand, Implement, and Use

REFERENCES

Apply Harvard referencing style. For example (<https://www.mendeley.com/guides/harvard-citation-guide/>):

Book Referencing Example:

Mitchell, J.A. and Thomson, M. (2017) *A guide to citation*. 3rd edn. London: London Publishings.

Journal Article Example

Mitchell, J.A. 'How citation changed the research world', *The Mendeley*, 62(9), p70-81.

Journal Article Online Example

Mitchell, J.A. 'How citation changed the research world', *The Mendeley*, 62(9) [online]. Available at: <https://www.mendeley.com/reference-management/reference-manager> (Accessed: 15 November 2016)

Website Example:

Mitchell, J.A. (2017). *How and when to reference* [Online]. Available at: <https://www.howandwhentoreference.com/> (Accessed: 27 May 2017).