# Assignment Brief 1 (RQF)

## Higher National Certificate/Diploma in Business

|  |  |
| --- | --- |
| **Student Name/ID Number:** |  |
| **Unit Number and Title:** | Unit 19: Data Structures and Algorithms |
| **Academic Year:** | **2021** |
| **Unit Assessor:** |  |
| **Assignment Title:** | Examine and specify ADT and DSA |
| **Issue Date:** |  |
| **Submission Date:** |  |
| **Internal Verifier Name:** |  |
| **Date:** |  |

|  |
| --- |
| **Submission Format:** |
| *Format:*   * The submission is in the form of an individual written report and a presentation. This should be written in a concise, formal business style using single spacing and font size 12. You are required to make use of headings, paragraphs and subsections as appropriate, and all work must be supported with research and referenced using the Harvard referencing system. Please also provide a bibliography using the Harvard referencing system.   *Submission*   * Students are compulsory to submit the assignment in due date and in a way requested by the Tutor. * The form of submission will be a soft copy posted on <http://cms.greenwich.edu.vn/>. * Remember to convert the word file into PDF file before the submission on CMS.   *Note:*   * The individual Assignment *must* be your own work, and not copied by or from another student. * If you use ideas, quotes or data (such as diagrams) from books, journals or other sources, you must reference your sources, using the Harvard style. * Make sure that you understand and follow the guidelines to avoid plagiarism. Failure to comply this requirement will result in a failed assignment. |
| **Unit Learning Outcomes:** |
| **LO1** Examine abstract data types, concrete data structures and algorithms  **LO2** Specify abstract data types and algorithms in a formal notation |
| **Assignment Brief and Guidance:** |
| **Assignment scenario**  You work as in-house software developer for Softnet Development Ltd, a software body-shop providing network provisioning solutions. Your company is part of a collaborative service provisioning development project and your company has won the contract to design and develop a middleware solution that will interface at the front-end to multiple computer provisioning interfaces including SOAP, HTTP, JML and CLI, and the back-end telecom provisioning network via CLI .  Your account manager has assigned you a special role that is to inform your team about designing and implementing abstract data types. You have been asked to create a presentation for all collaborating partners on how ADTs can be utilised to improve software design, development and testing. Further, you have been asked to write an introductory report for distribution to all partners on how to specify abstract data types and algorithms in a formal notation.  **Tasks**  **Part 1**  You will need to prepare a presentation on how to create a design specification for data structures, explaining the valid operations that can be carried out on the structures using the example of:   1. A stack ADT, a concrete data structure for a First In First out (FIFO) queue. 2. Two sorting algorithms. 3. Two network shortest path algorithms.   **Part 2**  You will need to provide a formal written report that includes the following:   1. Explanation on how to specify an abstract data type using the example of software stack. 2. Explanation of the advantages of encapsulation and information hiding when using an ADT. 3. Discussion of imperative ADTs with regard to object orientation. |

|  |  |  |
| --- | --- | --- |
| Learning Outcomes and Assessment Criteria (Assignment 1) | | |
| Pass | Merit | Distinction |
| **LO1** Examine abstract data types, concrete data structures and algorithms | | **D1** Analyse the operation, using illustrations, of two network shortest path algorithms, providing an example of each. |
| **P1** Create a design specification for data structures explaining the valid operations that can be carried out on the structures.  **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.  **M2** Compare the performance of two sorting algorithms. |
| **LO2** Specify abstract data types and algorithms in a formal notation | | **D2** Discuss the view that imperative ADTs are a basis for object orientation and, with justification, state whether you agree. |
| **P3** Using an imperative definition, specify the abstract data type for a software stack. | **M3** Examine the advantages of encapsulation and information hiding when using an ADT. |

1. Data structures
   1. Abstract data type (P1)
      1. Definition

Give definition of what is ADT, common ways to represent ADT

Hint:

1. *What is ADT?*
2. *How to define and implement ADT? (Base on 5 steps on Page 9 Slide 1 – describes more detail each step by yourseft – at least two sentences for each step)*
   * 1. Examples

Give examples of a general ADT, using a representation to illustrate. You can choose one of the following ADT: Singly Linked List, Doubly Linked List or Circular Linked List.

(The best way to illustrate operations of an ADT is using table)

*Hint:*

*You will choose only one of the above three types of Linked List and follow the question to introduce it :*

*1. define the chosen ADT: what it is? What is its data or attributes?*

*2. which supported operations and describe all them (in natual language) (explanation + illustration).*

* 1. ADT usages
     1. Application of Stack in memory (P2)

Describe an application of Stack in memory management: how the memory is organized, how a method (function) calls is implemented with stack.

*Hint:*

* *What is a stack? Definition, operation, exceptions of stack.*
* *How dose a memory stack work?*
* *What is a function call in a computer? How does it work?*
* *How can stack work in Function Calls? (shows: code, illustration and the detail explanation).*
  + 1. Application of an ADT (P3)

Describe a problem where you need an ADT (Array, Linked List, DLL, CLL), describe that data structure operations can be used for that problem.

*Hint:*

1. *Why did you choose the ADT in 1.1.2*
2. *What is an imperative ADT or imperative definition in ADT?*
3. *Using an imperative definition, specify your chosen ADT (all supported operations you mentioned in 1.1.2).*
4. *Use your* ***console*** *application to describe the problem.*
5. *Show the exact* ***effections what*** *all operation of your concrete class / ADT are called by the user.*

*(using table to describe)*

|  |  |  |
| --- | --- | --- |
| *Code in your main* | *Actual Output* | *Explanation* |
|  |  |  |

* + 1. Application of Queue (M1)

Describe a problem where you need Queue, describe a concrete queue for that problem and why using queue can solve that problem.

*Hint:*

1. *What is a queue?*
2. *Support operations (illustration and the detail explanation without code).*
3. *Queue exceptions and Pros and Cons of Queue*
4. *Describe a problem and why using queue to solve it.*
5. *Show your code to solve this problem by calling the Queue’s operations (Code and explanation)*
   1. ADT vs OOP
      1. Using OOP to represent ADT (M3)

Represent an ADT in Java programming language, explain how the ADT operations can be represented by methods of your class / interface.

*Hint:*

1. *What is Encapsulation and Information Hiding?*
2. *The differences between encapsulation and information hiding.*
3. *In Java, what do you use to represent an ADT and why?*
4. *When using an ADT, what is the role of Encapsulation and Information Hiding? (You can choose the ADT you mentioned in 1.1.2 to explain this section easily).* 
   * 1. Discussion (D2)

*Discuss the view that imperative ADTs are a basis for object orientation and, with justification, state whether you agree.*

1. Algorithms
   1. Sorting algorithms (M2)

*What is sorting algorithms? Classify them.*

* + 1. Bubble sort (or other O(n2) sort algorithms)

*Describe how it works (with illustration)*

*Analyze its performance (time, memory)*

* + 1. Quick sort (or other O(nlogn) sort algorithms)

*Describe how it works*

*Analyze its performance (time, memory)*

* 1. Shortest path algorithms (D1)

*What is the shortest path problem?*

* + 1. Dijkstra’s algorithm

*Describe how it works by using concrete example with illustration*

* + 1. Bellman Ford’s algorithm

*Describe how it works by using concrete example with illustration*