

ASSIGNMENT BRIEF

HTU Course No: 40201201	HTU Course Name: Data Structures & Algorithms
BTEC Unit Code: T/618/7430	BTEC UNIT Name: Data Structures & Algorithms



Student Name/ID Number/Section	
HTU Course Number and Title	40201201 Data Structures & Algorithms
BTEC Unit Code and Title	T/618/7430 Data Structures & Algorithms
Academic Year	2024-2025 2
Assignment Author	Balqis Aldabaibeh
Course Tutor	Balqis Aldabaibeh - Anas Amaireh - Fadia Alaeddin - Ahmed Bataineh - Reem Shtaiwi
Assignment Title	Implementing data structures and algorithms to solve real-world problems
Assignment Ref No	1
Issue Date	17/04/2025
Formative Assessment dates	From 17/04/2025 to 29/05/2025
Submission Date	15/06/2025
IV Name & Date	Abdullah Alamaren 16/04/2025
Submission Format	
<p>Part 1:</p> <ul style="list-style-type: none"> • Exam Type: In-class practical exam. • Exam Duration: 3 hours. • Exam Conditions: Closed Book. <p>Part 2:</p> <ul style="list-style-type: none"> • A full implemented and functioning code; for tasks that require implementation.. • Signed Declaration Form. • Report in .docx format. • An oral demonstration with your instructor of your submitted work <p>General Guidelines:</p> <ul style="list-style-type: none"> • The report should be well-formatted and in a Word version. Clearly state the answers by question part and subtask numbers. • All implemented ADTs and Algorithms should be fully implemented. • All submissions should be uploaded to the university LMS (the eLearning platform) within the stated time and date. • NO EMAIL SUBMISSIONS OR LATE SUBMISSIONS WILL BE ACCEPTED. • HTU policies and regulations will be applied for any kind of plagiarism. • The oral demonstration will include everything submitted in the written report, debugging and evaluating the submitted code, and the material basic concepts delivered during classes. • The attendance of the oral demonstration is mandatory, the exact date and time will be determined by your instructor and announced later . 	
Unit Learning Outcomes	
<p>LO1 Examine abstract data types, concrete data structures and algorithms</p> <p>LO2 Specify abstract data types and algorithms in a formal notation</p> <p>LO3 Implement complex data structures and algorithms</p> <p>LO4 Assess the effectiveness of data structures and algorithms.</p>	

Assignment Brief and Guidance

Part 1:

You are attending a job interview exam for software engineering position.

You should be familiar with the following concepts:

- Stack definition, memory stack and how it's used to implement function call.
- Asymptotic analysis, algorithm efficiency.
- Sorting algorithms.
- FIFO queue and it's variant implementation.

Part 2:

You are part of the IT support team at a university. Your team is developing a **Campus Management System** to support common student and staff services such as booking rooms, handling support requests, and guiding students around campus.

Your job is to design and implement specific features of this system using the most appropriate data structures and algorithms.

- **Feature 1- Track lost/found record:** track lost and found items stored with description, date, and location.
- **Feature 2- Room Booking:** students request bookings private rooms for study, the processing of requests depends on some priority.
- **Feature 3- Student Orientation Help:** guide new students to navigate between buildings and find the shortest path between two locations in campus.
- **Feature 4- Undo/Redo:** Allow users to undo or redo their last action in campus system, ex. room booking.
- **Feature 5- Event Participation Registration:** register students for events and track registrations in the order received.
- **Feature 6- Student Locker Allocation:** assign lockers on student ID and allow quick lookup, remove or register lockers as students graduate or withdraw.

1. Provide a design specification using class diagram for the used data structures, explaining their valid operations. The design should consider a set of different ADTs for each single feature. Naming; List, Stack, Queue, Trees(BST), and Graph.
2. Implement the provided solution using JAVA programming language, provided with a user menu, where user can select any option to execute. In addition to provide error handling cases where needed. Explain with use cases how your implementation solves the given problem, and justify your decision of ADTs and demonstrate how the system meets basic user needs. Evaluate three benefits of using the selected independent ADTs.
3. Critical evaluate the complexity of each implemented algorithm and ADT in the proposed solution.
4. For feature #3, illustrate step by step in details the implemented algorithms.
5. In reference to your solution, explain how information hiding and encapsulation helped you when using the implemented ADTs. Discuss with justification the view that ADTs are a basis for OOP.
6. For each implemented feature using a selected ADT, interpret the trade-off with its alternatives in terms of time and space complexities for all valid operations, support your answer with examples. For example, if LinkedList ADT were chosen to implement feature #1, compare it with other possible alternative ADTs to implement same feature such as ArrayList in terms of insert, remove, ..etc.

Learning Outcomes and Assessment Criteria			
Learning Outcome	Pass	Merit	Distinction
LO1 Examine abstract data types, concrete data structures and algorithms	<p>P1 Create a design specification for data structures, explaining the valid operations that can be carried out on the structures.</p> <p>P2 Determine the operations of a memory stack and how it is used to implement function calls in a computer.</p>	<p>M1 Illustrate, with an example, a concrete data structure for a First in First out (FIFO) queue.</p> <p>M2 Compare the performance of two sorting algorithms.</p>	<p>D1 Analyse the operation, using illustrations, of two network shortest path algorithms, providing an example of each.</p>
LO2 Specify abstract data types and algorithms in a formal notation	<p>P3 Specify the abstract data type for a software stack using an imperative definition.</p>	<p>M3 Examine the advantages of encapsulation and information hiding when using an ADT.</p>	<p>D2 Discuss the view that imperative ADTs are a basis for object orientation offering a justification for the view.</p>
LO3 Implement complex data structures and algorithms	<p>P4 Implement a complex ADT and algorithm in an executable programming language to solve a well-defined problem.</p> <p>P5 Implement error handling and report test results.</p>	<p>M4 Demonstrate how the implementation of an ADT/algorithm solves a well-defined problem.</p>	<p>D3 Critically evaluate the complexity of an implemented ADT/algorithm.</p>
LO4 Assess the effectiveness of data structures and algorithms.	<p>P6 Discuss how asymptotic analysis can be used to assess the effectiveness of an algorithm.</p> <p>P7 Determine two ways in which the efficiency of an algorithm can be measured, illustrating your answer with an example.</p>	<p>M5 Interpret what a trade-off is when specifying an ADT, using an example to support your answer.</p>	<p>D4 Evaluate three benefits of using implementation independent data structures.</p>