

Curriculum Area	Awarding Body
Digital & Creative Innovation - BSoC	Lancaster University

Assessment Details			
No	2 of 2	Type	Coursework - Letter Grade
Word count	2000	Weighting	50%

Module and Programme Details			
Title	Data Structures and Algorithms		
Credits	20	Stage	Stage 1 Level 4
Code	DN4MD002	Occurrence	25/26
	DN1WD001 25/28 BSc (Hons) Digital and Technology Solutions - Software Engineer		

Tutor Details			
Name	Dr Graham Burvill	Email	Graham.burvill@blackpool.ac.uk
Room	SB052A	Tel	01253 50414

Internal Verification	
IV Name	Michael Thornhill
Date	9 th January 2026

Distribution	
Date	[W/C] 5th January 2026

Submission and Feedback			
Location	Canvas upload	Feedback W/C	23rd March 2026
Date	3 rd March 2026	Time	12:00 Noon

Please note that late submissions may incur a penalty as defined within the assessment regulations of the awarding body

If you would like support from a [HELM](#) please complete the referral form [here](#)

Student/Apprentice Declaration

It is the responsibility of each student to ensure that their work is submitted in the correct format, that the appropriate files have been uploaded, and that all materials intended for assessment are included. Upon completing the online submission, students are advised to download the submitted file to verify that it has been received as intended. Please note that if the College is unable to open the submitted file, it may not be eligible for marking.

The College is committed to ensuring, as far as reasonably possible, that all work submitted by students and apprentices is their own, and that academic credit is not awarded for unreferenced material originating from external sources, including content generated by artificial intelligence.

By submitting work for this assignment, you confirm that the content is entirely your own and has not been submitted, in whole or in part, for the award of any degree or qualification by yourself or any other individual. You also affirm that all sources have been appropriately cited and referenced.

Students are expected to be familiar with the College's regulations concerning assessment and academic malpractice. Please be aware that all submissions will be subject to automated checks for plagiarism and the use of AI-generated material.

Learning Outcomes Assessed in the Assignment

3. Design algorithms using software engineering methods.
4. Develop an application to manage data.
5. Test and optimise developed algorithms.

Grade Descriptors (Aligned to the learning outcomes and the primary descriptors of the awarding body) <i>The descriptions below provide an indication of the requirements for each grade boundary for this assignment</i>	
Excellent	<p>Demonstrates excellent, well-structured, and justified algorithm design using appropriate software engineering methods</p> <p>Application is a fully functional development that is logically structured and effectively manages data which shows excellent alignment with the design.</p> <p>Testing is thorough and well-documented, showing excellent and clear links to algorithm behaviour, with evidence of debugging and optimisation.</p>
Good	<p>Demonstrates good, structured algorithm design with justification, using some appropriate software engineering methods</p> <p>Application functions as intended which is structured and reflects good alignment with intended design.</p> <p>Testing is appropriate and documented, with good evidence of debugging and at least one improvement applied.</p>
Satisfactory	<p>Provides a satisfactory basic algorithm design with limited detail or justification.</p> <p>Application functions at a basic and satisfactory level but lacks robustness.</p> <p>Testing is present and satisfactory but is but limited in scope or clarity.</p>
Weak	<p>Algorithm design is weak and shows unclear, incomplete, or poorly justified design.</p> <p>Application is weak as it shows significant issues or incomplete functionality. Testing evidence is weak or poorly documented.</p> <p>Limited or no meaningful optimisation demonstrated. Writing lacks clarity, structure, or accurate referencing.</p>
Fail	<p>The solution proposed fails to cover or is deficient in:</p> <ul style="list-style-type: none"> • Structure and justification of Algorithm design • Providing a working application to manage data.

- Testing and optimising an application a developed algorithm

Additional Requirements (where applicable)

This section describes any additional assessment requirements which may be identified by public statutory and regulatory bodies

K26: How to select and apply a range of software tools used in Software Engineering.

S19: Implement software engineering projects using appropriate software engineering methods, approaches and techniques.

Assignment Brief

SCENARIO

Following a successful investigative stage on modernising their “Task Tracker” tool, DataCapture a small digital services company has asked you to extend and formalise the Task Tracker system.

While the original proof of concept demonstrated appropriate data structure selection, the company now requires clearer algorithm design, improved application structure and evidence of testing and refinement

You are required to develop and test a small application that manages task data using clearly defined algorithms and structured software engineering practice.

You are required to:

- Design one or more algorithms that support the operation of the Task Tracker system
- Implement these algorithms in a working application
- Test the algorithms systematically
- Identify and apply at least one improvement or optimisation

Your submission must demonstrate how your design decisions influence your implementation and testing and reflect professional software engineering practice.

REPORT

Your report should be approximately 2,000 words ($\pm 10\%$) and include the following sections:

1. Algorithm Design

Describe the key algorithms used in your system (e.g. task retrieval, priority selection, update logic)

Use appropriate design methods such as:

- pseudocode
- flow diagrams
- structured step-by-step descriptions

This should clearly explain any inputs, decision points, and outputs

2. Application Development

Describe how your algorithms were implemented in code and explain how the application manages data.

This should justify your choice of:

- Programming languages, tools and environment used during development
- Demonstrate clear alignment between design and implementation

3. Testing Strategy

Describe your approach to testing including any test cases, expected vs actual outcomes. You should

explain how testing helped identify errors, issues or improvements

Include evidence of debugging where appropriate

4. Optimisation and Improvement

Identify at least one limitation or inefficiency in your original implementation. You should describe the improvement or optimisation applied and explain what was changed, why it improves the system and any impact on maintainability or performance

5. Reflection

Your reflection should discuss what worked well, any challenges faced and ways to improve with more time and how this work prepares you for future software development tasks

APPLICATION BUILD

You must submit a working application that implements the designed algorithms, manages task data effectively, and reflects the design decisions as discussed in your report

Your code should:

- Be clearly structured
- Include meaningful comments
- Match the behaviour described in your report

Submit all code files in either in a single .zip file or in a Github repository.

DELIVERABLES

Your submission must include:

- A 2,000-word report ($\pm 10\%$)
- A ZIP file containing all source code or a link to GitHub embedded in your report.

Both items must be submitted via Canvas by the stated deadline.

*The referencing should all be undertaken in the **Harvard Referencing format**, and any images used should be used in line with the text throughout the document (with each image being given a figure number and caption).*