



## ASSESSMENT BRIEF

<b>Module Title:</b>	Algorithms and Data Structures
<b>Module Code:</b>	KV5033
<b>Academic Year / Semester:</b>	2024-25 / Semester 1
<b>Module Tutor / Email (all queries):</b>	Nick Dalton Nick.dalton@northumbria.ac.uk
<b>% Weighting (to overall module):</b>	50%
<b>Assessment Title:</b>	Assessment #1
<b>Date of Handout to Students:</b>	TBD
<b>Mechanism for Handout:</b>	Module Blackboard Site & Seminar in Week 5
<b>Deadline for Attempt Submission by Students:</b>	17/Jan/2025
<b>Mechanism for Submission:</b>	Document upload to Module Blackboard Site
<b>Submission Format / Word Count</b>	Online - code
<b>Date by which Work, Feedback and Marks will be returned:</b>	TBD
<b>Mechanism for return of Feedback and Marks:</b>	Mark and individual written feedback sheet will be uploaded to the Module Site on Blackboard. For further queries please email module tutor.

### LEARNING OUTCOMES

The learning outcomes (LOs) for this module are:-

**LO1** Knowledge & Understanding: Demonstrate an understanding of the basic principles of programming using an appropriate programming language, including the use of variables, conditions, loops, subprograms

**LO2** Demonstrate an understanding of abstraction mechanisms and structured data types

**LO3** Intellectual / Professional skills & abilities: Create reliable and maintainable software using appropriate code standards.

**LO4** Personal Values Attributes (Global / Cultural awareness, Ethics, Curiosity) (PVA): Solve problems using a structured approach.

**This assessment addresses learning outcomes LO1, LO2, LO3 and LO4.**



**Nature of the submission required:**

See the section how to submit work.

**Instructions to students:**

*This is an individual piece of work.*

**Referencing Style:**

N/A

**Expected size of the submission:**

Just the source code.

**Academic Conduct:**

You must adhere to the university regulations on academic conduct. Formal inquiry proceedings will be instigated if there is any suspicion of misconduct or plagiarism in your work. Refer to the University's regulations on assessment if you are unclear as to the meaning of these terms. The latest copy is available on the university website.



## What to do

Download the assessments from Blackboard

## How to submit work

Upload your code and answers to Blackboard -

If for some reason this does not work, zip Assessment2.py with your name e.g JoeDoe.zip.

Experimental Invite the tutors (@sheepdalton) to your fork on GitHub

## Questions

This consists of 3 mini projects downloaded f

The projects grow in difficulty and complexity. If for some reason your code will not work. You can give flow charts to show how you think the program could be speeded up. These get a reduced mark ( up to 80% of mark ) which is better than no code and no mark.

1. Project 1 (30%) – you are required to read in a file, process it then save the result. This is similar to what was done in Workshop 7. Choice of why the code is slowed down.
  - a. You must provide the profile of the old version.
  - b. You must identify the function which is genuinely slowing the program down.
  - c. You must also identify the o-notation for the function.
  - d. You must change the code to speed it up.
  - e. You must identify the final o-notation.
  - f. You must run a profile on the new version.
2. Project 2 (30%) you are required to process some generated data and then compute the results. This has a more complex problem.
  - a. You must provide the profile of the old version.
  - b. You must identify the function which is genuinely slowing the program down.
  - c. You must also identify the o-notation for the function.
  - d. You must change the code to speed it up.
  - e. You must identify the final o-notation.
  - f. You must run a profile on the new version.
3. Project 3 (40%) you are required to process some generated data and then compute the results. Slightly more complex than previous version.
  - a. You must provide the profile of the old version.
  - b. You must identify the function which is genuinely slowing the program down.
  - c. You must also identify the o-notation for the function.
  - d. You must change the code to speed it up.



**Northumbria  
University**  
NEWCASTLE

- e. You must identify the final o-notation.
- f. You must run a profile on the new version.