Northeastern University London

Assessment Brief: Group Work with Essay 2024–25

Assessment Details

Course Title:	Fundamentals of Computer Science II
Course Code:	LCSCI4208
Course Leader:	Fatemeh Parsa
Level:	4
First, Second, or Third Sitting:	First
Assessment Title:	Project
Assessment Number:	AE3
Assessment Type:	Code & Report / Documentation
Restrictions on Time/Length:	24–32 hours
Assessment Weighting:	30%
Issue Date:	13 March 2025
Hand-in Deadline:	10 April 2025, at 13:00
Mode of Submission:	Online (Canvas)
File Format Accepted:	.zip file with archived Maven project
Planned Feedback Deadline:	28 calendar days after hand-in deadline or last presentation day, whichever later
Anonymous Marking:	NO

Assessment Task

For your second project, you will continue your work from AE2 on image compression. Working in groups of two to three students, you will extend the program to remove vertical seams instead of columns.

There is a starter code available for this project, and your task is to complete the missing parts of the provided code, ensuring correctness and functionality. The provided tests should be used to validate that your implementation works correctly.

A vertical seam is a sequence of pixels extending from the top to the bottom of an image, where each pixel is directly or diagonally below the previous pixel in the sequence. Your task is to remove either the greenest seam (the seam with the highest cumulative green value) or the seam with the lowest energy in the image.

The seam with the lowest energy will minimally impact the recognizability of the image upon removal.

We define energy using the following formula, given that br(pixel) is the brightness of the pixel. Brightness is calculated as the average of the RGB values for the given pixel.

Consider pixel E below, which has eight neighbors: A, B, C, D, F, G, H, and I.

A	B	C
D	**E**	F
G	H	I

We define horizontal and vertical energies as follows:

$HorizEnergy(E) = (br(A) + 2br(D) + br(G)) - (br(C) + 2br(F) + br(I))$

$VertEnergy(E) = (br(A) + 2br(B) + br(C)) - (br(G) + 2br(H) + br(I))$

Thus, the energy at E is:

Energy(E) = $\sqrt{HorizEnergy(E)^2 + VertEnergy(E)^2}$

Seam Finding

To find the lowest-energy seam, an inefficient method would be to compute all possible seams and select the smallest. This approach, however, would significantly slow down the program, requiring both extensive computation and storage. A more efficient solution recognizes the overlap between seams and utilizes a dynamic

programming table to store energies and determine the minimum seam by comparing neighboring values.

Consider the following example pixel energies for a 4x4 image:

5	6	3	8
4	1	6	4
3	2	1	3
8	6	5	2

The first row (5 6 3 8) has no neighbors above it, making each initial value the minimum at its respective position (See note 2 for further details on edge pixels). Next, consider the pixel valued 4 in the second row, first column. Its neighbors above are 5 and 6. Summing these with the current pixel yields:

- 4 + 5 = 9
- 4 + 6 = 10 The minimum is 9, coming from the pixel valued 5.

Repeat this for each pixel, moving row by row and choosing the smallest cumulative energy. Continue until all rows are processed as follows:

Consider the 4 (bold, shaded) in the table below:

5	6	3	8
4	1	6	4
3	2	1	3
8	6	5	2

It has 5 and 6 as neighbors above. If we were to sum 4 and 5 we get 9, 4 and 6, 10. Since 9 is smaller that is the new value for the path and we record that it came from 5.

5	6	3	8
9 (above)	1	6	4
3	2	1	3
8	6	5	2

Now consider the pixel in the second row and second column, it is a 1. Its neighbors above are 5, 6, and 3:

5	6	3	8
9 (above)	**1**	6	4
3	2	1	3
8	6	5	2

The smallest sum is 3 and 1, so the value becomes 4 from 3 above and to the right.

5	6	3	8
9 (above)	**4** **(above,right)**	6	4
3	2	1	3
8	6	5	2

Continuing the row becomes:

5	6	3	8
9 (above)	4 (above,right)	9 (above)	7 (above,left)
3	2	1	3
8	6	5	2

Repeating with the next row:

5	6	3	8
9 (above)	4 (above,right)	9 (above)	7 (above,left)
7 (above,right)	6 (above)	5 (above,left)	10 (above)
8	6	5	2

Finally, last row:

5	6	3	8
9 (above)	4 (above, right)	9 (above)	7 (above, left)
7 (above, right)	6 (above)	5 (above,left)	10 (above)
14 (above, right)	11 (above, right)	10 (above)	7 (above, left)

Now tracing it backward from 7, we see 3, 4, 5, 7 as the lowest energy seam. These values are the smallest within each row, that is, 3 is the smallest at row 1, 4 is the smallest at row 2, 5 the smallest at row 3 and 7 the smallest at row 4. The resulting seam, in the table below, is the path (top-bottom) following the cells with thick borders.

5	6	**3**	8
9 (above)	**4** **(above, right)**	9 (above)	7 (above,left)
7 (above, right)	6 (above)	**5** **(above, left)**	10 (above)
14 (above, right)	11 (above, right)	10 (above)	**7** **(above, left)**

Note: You must also implement the functionality to reinsert seams in reverse order of removal, providing an "undo" capability. Therefore, track removed seams efficiently, along with the exact locations for reinsertion.

Note 2: When computing the energy, pixels located on the boundary (edges) of the image do not require energy calculation. Instead, you should use their brightness, as these boundary pixels lack the necessary full set of neighbors for the standard energy formula.

User Interface

Your program should respond to the following key inputs:

- "g": Display the greenest seam (highlighted in green) and remove it.
- "e": Display the lowest-energy seam (highlighted in red) and remove it.
- "u": Undo the deletion by reinserting the most recently removed seam.
- “q”: Quit the program.

Users should be able to remove and restore multiple seams consecutively.

Time Restrictions

Your program must include methods to remove and reinsert seams, each running in O(n) time, where n is the image height in pixels. Therefore, a standard 2D array or ArrayList is insufficient. Consider using a graph structure.

Note: Updating pixel energies after seam removal or insertion operations will require $O(n^2)$ time.

Displaying Images

Your program should export changes at each step:

- Greenest seam selection: Export image with seam highlighted in green.
- Lowest-energy seam selection: Export image with seam highlighted in red.
- Seam deletion: Export image with seam removed.
- Undo operation: Export image after reinserting the most recently removed seam.

Summary

In this project, you will:

1. Write a program to compress images by removing seams rather than columns. Define seams as pixel sequences starting from the top row and selecting each subsequent pixel from the three available pixels below (down-left, down, down-right) based on either the highest green value or the lowest energy, continuing to the bottom row.
2. Allow users to remove either the greenest seam or the lowest-energy seam, highlighting the selected seam in green or red, respectively.
3. Develop a user interface enabling users to choose between greenest or lowest-energy seam deletion, confirm deletions, and perform undo operations.
4. Export the resulting images after each operation.

Assessment Criteria

GROUP WORK (60 Marks)	
Seam removal implementation	**25 Marks**
• Correct identification of vertical seams	5 Marks
• Correct calculation of seam energy	5 Marks
• Efficient dynamic programming algorithm implementation	5 Marks
• Correct removal of lowest-energy seam	5 Marks
• Correct removal of greenest seam	5 Marks
Undo functionality	**10 Marks**
• Efficient tracking of removed seams	5 Marks
• Correct reinsertion of seams (undo feature)	5 Marks
User interface	**15 Marks**
• Implementation of seam highlighting (color accuracy)	5 Marks
• Functionality of key inputs ("g", "e", "u", “q”)	5 Marks
• Usability and responsiveness of the interface	5 Marks
Image export and visualization	**10 Marks**
• Accurate exporting after each operation	5 Marks
• Visual clarity and correctness of exported images	5 Marks
INDIVIDUAL REPORT (40 Marks)	
Technical explanation	**20 Marks**
• Clear explanation of seam identification and removal process	5 Marks
• Accurate description of energy calculation	5 Marks
• Discussion of efficiency and data structures used (graph vs array)	5 Marks
• Clear explanation of undo implementation	5 Marks
Evaluation and analysis	**10 Marks**
• Evaluation of algorithm efficiency and performance	5 Marks
• Reflection on challenges and solutions	5 Marks

Presentation and clarity	**10 Marks**
• Organization, and readability of report	5 Marks
• Clarity and correct use of language	5 Marks

70 or higher There was evidence of the ability to perform all programming tasks correctly. The demonstration of the methods was excellent, coherent, well documented and clearly explained.

60-69 There was evidence of ability to perform some programming tasks correctly. The demonstration of the methods is good, coherent and reasonably detailed and explained.

50-59 There was evidence of ability to perform some programming tasks correctly, but the demonstration of the methods was limited, incoherent, not adequately documented and vaguely explained.

40-49 There was limited evidence of ability to perform programming tasks. The demonstration of the methods involved significant omissions and produced substantial inaccuracies.

39 or less Failure to solve the programming task in assignment. Methods were completely incorrect or absent. General grading criteria for Level 4 are described in Appendix B of the course syllabus.

Submitting Assessments

All students must submit the entire group work submission as well as their individual work. The submitted group work must be identical between all group members.

You must submit your work as a ZIP file to **Canvas**. This file should:

- Preserve the directory structure covered during the course (i.e., main, test, package)
- Preserve the directory structure covered during the course (i.e., main, test, package, pom.xml). This should be a valid Maven project.
- Include all of the corresponding Java files that perform the tasks requested above, that is, not just the class containing the `main` method.
- Comprise the image(s), in PNG format, involved in the project:
    - The original image
    - At least 1 image resulting from applying the *greenest seam* to the original image.

    - At least 1 image resulting from applying the *lowest energy seam* to the original image.
- Put comments on what the code does in the Java source code.

You have three submission attempts, but only the last submission will be graded. If your last submission attempt is late, you will receive the late penalty even if you have a previous submission that was on time. Please make sure to avoid multiple submissions for assessments with multiple components, as only the last attempt will be graded. Upload several files in one submission attempt instead.

If your assessment requires anonymous submission (see the assessment details table at the top of your assessment brief), please be sure you have left your name off of your submission and out of the submission file name, as failing to do so may result in a 0% mark on the assessment.

Refer to the assessment details table in your assignment brief for acceptable file formats. Avoid submitting zip files (unless explicitly required by the assessment brief); use the 'add files' function to submit multiple files instead. If you are submitting a physical artefact, you must also provide clear and thorough documentation (such as in the form of photographs or a video) of your submission by the deadline; see the bottom of this section for guidance on submitting video files.

Please ensure that you tick the agreement box at the very bottom of your Canvas submission page (scroll down if you don't see it). This will enable you to select 'Submit Assessment.' Please review the submitted file to ensure that everything is in order.

If you encounter any issues with submission, e-mail a copy of your assignment before the deadline to student.assessments@nulondon.ac.uk along with screenshots of the problem on Canvas, showing a timestamp.

To turn on notifications for submission confirmation emails in your Canvas settings: Account > Notifications > Turn on the bell for 'All submissions.' In the app this is via Settings > Email Notifications > All submissions.

To submit a video recording: Select the 'Panopto video' icon in the text entry box in your submission portal. You can upload a video file of any format from your media library by selecting 'upload,' choosing 'my folder' in the drop down menu, and clicking 'insert.' You should be able to play the video back once it processes. See further explanation, including guidance on recording videos using Panopto, in this support article: 'How to Submit a Video Assignment in Canvas.'

Marking

The University uses two categorical marking schemes – one for undergraduate and one for postgraduate – to mark all taught programmes leading to an award of the University.

More detailed information on the categorical marking scheme and the criteria can be found in the Course Syllabus, available on the University's VLE.

Learning Outcomes

This assessment will enable students to demonstrate in full or in part the learning outcomes identified in the Course Descriptor.

On successful completion of this assessment, students should be able to:

Knowledge and Understanding

K1a Demonstrate knowledge and understanding of basic design principles, programming practices, tools, and techniques for software development.

K3a Choose among basic design patterns and practices to design and implement small to moderately sized programs.

Subject-Specific Skills

S1a Evaluate the impact of program design decisions and implementations on the technical, social and management dimensions of software.

S2a Demonstrate practical use of object-oriented design concepts, including encapsulation and inheritance, in software.

Transferable Skills

T1a Write clear, concise, and well-documented code.

T3a Display a developing technical proficiency in written English and an ability to communicate clearly and accurately in structured and coherent pieces of writing.

Accessing Feedback

Students can expect to receive feedback on all summative coursework within 28 calendar days of the submission deadline or, if applicable, the last oral assessment date, whichever later. The 28 calendar day deadline does not apply to work submitted late. Feedback can be accessed through the assessment link on the Canvas course page.

Late Submissions

Please ensure that you submit your assignment well before the deadline to avoid any late penalties, as a submission made exactly on the deadline will be considered late. Please keep in mind that there may be differences between your computer's clock and the server time, which can cause discrepancies, and that Canvas may take some time to process your submission. All group members are responsible for the timely submission of the assessment.

Your Canvas submission portal displays two due dates: one is the deadline for your assignment, and the second is the latest possible date by which your assignment can be submitted late. Please make sure you submit by the assessment deadline in order to avoid late penalties.

If assessments are submitted late without approved Extenuating Circumstances, there are penalties:

- For assessments submitted up to one day late, any passing mark will receive 10 marks deducted or a threshold pass (40% for undergraduate students, 50% for postgraduate students), whichever is higher. Any mark below 40% for undergraduate students and below 50% for postgraduate students will stand.
- Students who do not submit their assessment within one day of the deadline, and have no approved Extenuating Circumstances, are deemed not to have submitted and to have failed that assessment element. The mark recorded will be 0%.

For further information, please refer to AQF7 Part C in the Academic Handbook.

Extenuating Circumstances

The University's Extenuating Circumstances (ECs) procedure is in place if there are genuine circumstances that may prevent a student from submitting an assessment. If the EC application is successful, there will be no academic penalty for missing the published submission deadline.

Students are normally expected to apply for ECs in advance of the assessment deadline. Students may apply for consideration of ECs retrospectively if they can provide evidence that they could not have done so in advance of the deadline. All applications for ECs must be supported by independent evidence.

Students are reminded that the ECs procedure covers only short-term issues (within 21 days leading to the submission deadline) and that if they experience longer-term matters that impact on learning then they must contact Student Support and Development for advice.

Under the Extenuating Circumstances Policy, students may defer an assessed element on only one occasion and may request an extension on a maximum of two occasions.

For further information, please refer to the Extenuating Circumstances Policy in the Academic Handbook.

Academic Misconduct

Any submission must be a student's own work and, where facts or ideas have been used from other sources, these sources must be appropriately referenced. The University reserves the right to hold a viva if there are concerns about the authenticity of a student's or learner's work. The Academic Misconduct Policy includes the definitions of all practices that will be deemed to constitute academic misconduct. This includes the use of artificial intelligence (AI) where not expressly permitted within the assessment brief, or in a manner other than specified. Students should check this policy before submitting their work. Students suspected of committing Academic Misconduct will face action under the Policy. Where students are found to have committed an offence they will be subject to sanction, which may include failing an assessment, failing a course or being dismissed from the University depending upon the severity of the offence committed. For further information, please refer to the Academic Misconduct Policy in the Academic Handbook.

Version History

Title: Assessment Brief: Oral Summative Assessment **Approved by: The Quality Team**					
Version number	**Date approved**	**Date published**	**Author**	**Location**	**Proposed next review date**
4.0	March 2025				
3.0	March 2023	March 2023	Registrar	VLE/ Faculty Resources Page	March 2024
2.0	August 2022	August 2022	Registrar	VLE/ Faculty Resources Page	July 2023
1.1	December 2021	December 2021	Registrar	VLE	August 2022
1.0	August 2021	August 2021	Registrar	VLE	August 2022
Referenced documents	AQF7 Academic Regulations for Taught Awards; Extenuating Circumstances Policy; Academic Misconduct Policy; Course Syllabus				
External Reference Point(s)	UK Quality Code Theme: Assessment				