ECS759P Artificial Intelligence
Coursework 1

This coursework contributes **50% of your final module grade**.

Deadline: 31st of October 2025, at 10AM, on QM+

Communication and Support: If you have questions about this coursework, please post them on the **Student Forum on QMplus**. Using the forum allows all students to benefit from clarifications and official responses. You may also ask questions during your assigned lab sessions. For individual inquiries, you can contact **Dr. Iran R. Roman** during **office hours (Friday, 10–11am, Peter Landin 4.16)** or by email at i.roman@qmul.ac.uk.

Submission Instructions: Your final submission **must include the following files**:

- `report.pdf` — containing all required written answers, explanations, and figures
- `cw1.py` — containing the code implementations specified in this coursework assignment.

If you reuse code from the lab sessions, you must also include the corresponding `.py` lab files (e.g. `lab1.py`, `lab2.py`, etc.) in your submission directory. IMPORTANT: do NOT submit jupyter notebooks (DO NOT submit files with extension `.ipynb`). We will not look at those at all. Failure to include all required files, correctly named and in the proper format, will result in your submission being automatically flagged as **INCOMPLETE** and an automatic score of ZERO.

Your complete submission should be within a single `.zip` file.

Coding Guidelines:

- You **must not use external libraries** beyond those used in the labs.
- Your code should be clear, well-documented, and modular.
- All required functions must be defined using the exact names and signatures specified in the starter code in `cw1.py`

Report Guidelines:

- Your `report.pdf` should include all required written answers, figures, and explanations.
- Answers must be clearly labeled to correspond to the correct question number.
- You may use any typesetting software (e.g. LaTeX, Overleaf, or Word). Using LaTeX is recommended.
- If you prefer to write answers by hand, take clear, legible photos of your work and combine them into a single PDF file named `report.pdf`.

Coursework structure: 6 main "Questions" (total 100 marks):

1. **Four Categories of AI** — 10 marks

2. **Research Paper Analysis** — 5 marks

3. **Jay McClelland & AI History** — 20 marks

4. **Agent Description** — 5 marks

5. **Path Finding Warmup** — 10 marks

6. **Networks and Pathfinding (including heuristic search)** — 50 marks

Assessment Criteria:

- **Correctness:** Does your code produce accurate and reliable results?
- **Design Quality:** Are your algorithms well-structured, efficient, and logically sound?
- **Clarity:** Are your explanations and justifications concise, coherent, and technically clear?
- **Presentation:** Is your report professionally formatted, well-organized, and easy to read?

Academic Integrity: All submitted work must be your own. Collaboration on conceptual understanding is encouraged, but direct sharing of code or written text is strictly prohibited. Any detected plagiarism or code similarity beyond what is expected for independent work will be reported and handled according to the **Queen Mary Academic Misconduct Policy**.

IMPORTANT NOTE ABOUT YOUR USE OF AI AND LLMs

The use of Generative Artificial Intelligence tools (such as **ChatGPT**, **Claude**, **GitHub Copilot**, **Gemini**, or other large language models, LLMs) is **permitted and encouraged** in this coursework, provided that it is used **critically, transparently, and responsibly**.

If you use AI tools in any part of your work (including code or report writing), you must:

- Mention FULL **name and version** of the tool (e.g., "ChatGPT, GPT-5, October 2025").
- Add a brief **note or comment** in your report or code indicating which parts were assisted or generated by AI.
- At the end of your report, summarize in less than 200 words **how you used the AI tool** (for example: to rephrase text, debug code, explore alternative ideas, your prompting style/technique).

You do **not** need to include every individual prompt or an extensive log of interactions. A short, honest statement of use is enough. The goal is **transparency**, not extra workload. **Full transparency** is required to receive credit for AI-assisted work. Using AI without clear attribution or explanation will be treated as academic misconduct and will result zero.

While the use of AI is allowed, you are still responsible for the **accuracy, originality, and quality** of all submitted content. The ability to critically evaluate, adapt, and improve AI-generated outputs will be considered a mark of strong performance.

Remember: AI is a tool, not a substitute for your understanding. Use it to enhance, not replace, your own reasoning and creativity.

1 Four Categories of AI (Total marks: 10)

For each of the "four categories of AI" discussed in Lecture 1, complete the following tasks:

a) Create one prompt on any topic that would cause an AI system to respond in a way matching that category

b) Use a real AI system with your prompt

c) Record the exact, unedited response from the AI

d) State which AI system and version you used

e) Write a short explanation addressing:

   - Why your prompt fits the category
   - Which parts of the response demonstrate that category

Submission Format

Repeat this format for each of the four categories:

- **Category:** [Category name]
- **Prompt:** [Your question or instruction]
- **AI Agent + Version:** [e.g., ChatGPT Free (GPT-3.5)]
- **Response:** [Unedited AI reply]
- **Explanation:** [Your reasoning]

Requirements: maximum 30 words per explanation.

2 Research Paper Analysis (Total marks: 5)

Read the paper at tinyurl.com/2s5nyt73 (written by a Queen Mary MSc student) and answer the following:

Part a) [1 mark]

According to the paper, what is the significance of “Pain” in the context of AI?

Requirements: maximum 20 words.

Part b) [3 marks]

Considering the “categories of AI,” which category would you use to describe the author’s model? Explain your rationale.

Requirements: maximum 35 words

Part c) [1 mark]

Describe one aspect of the methodology that could be easily changed to enhance or expand the study.

Requirements: maximum 20 words

3 Jay McClelland & AI History (Total marks: 20)

Jay McClelland, a professor at Stanford University, has been a leading figure throughout AI's history. This question explores his intellectual journey and connections to key moments in AI development.

Part a) Historical Timeline [10 marks: 1 mark per event]

Write a paragraph where, for each historical event below, you describe where McClelland was at that time and his involvement or connection to the event.

Requirements: Maximum 150 words total for all events combined.

Events:

1. **1950** – Alan Turing publishes *Computing Machinery and Intelligence* in Mind
2. **1957** – Frank Rosenblatt presents the Perceptron model at Cornell Aeronautical Laboratory
3. **1969** – Minsky and Papert publish *Perceptrons*, critiquing neural network capabilities
4. **1980** – John Searle introduces the Chinese Room argument challenging strong AI claims
5. **1986** – Rumelhart, Hinton, and Williams publish *Learning Representations by Back-Propagating Errors*
6. **1997** – IBM's Deep Blue defeats world chess champion Garry Kasparov
7. **2006** – Geoffrey Hinton and collaborators publish *Reducing the Dimensionality of Data with Neural Networks*
8. **2012** – Krizhevsky, Sutskever, and Hinton release AlexNet, winning the ImageNet competition
9. **2017** – Vaswani et al. publish *Attention Is All You Need*
10. **2024** – OpenAI releases GPT-4 and Google DeepMind introduces Gemini

Part b) Foundations of AI [4 marks]

Relate McClelland to two of the "Foundations of Artificial Intelligence" from Chapter 1 of the Russell and Norvig textbook.

Requirements: Maximum 40 words total

Part c) Network Positioning [2 marks]

Revisit the "Network" slide from Lecture 1 (slide "70"). On which side of the network would you place McClelland, and why?

Requirements: maximum 20 words

Part d) Bibliography and Citations [4 marks: 1 mark per source]

Include a bibliography with four highly relevant peer-reviewed papers from reputable sources. Cite these sources throughout your answers to this Question 3.

Requirements:

- Exactly 4 peer-reviewed papers
- All 4 must be cited in your answers this Question 3
- If more than 4 sources are included, only the first 4 will be marked

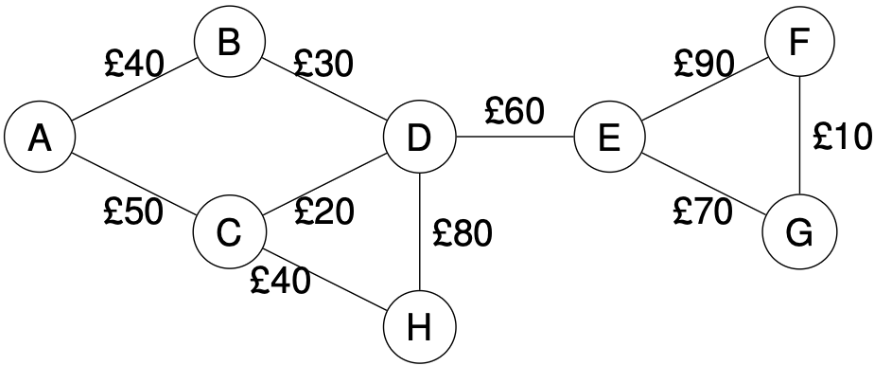
4 Agent Description (Total marks: 5)

Recall the “Agent Description Schema” from Lecture 2, and the presentation by the guest speaker that same day. Define the guest speaker’s agent(s) based on the “Agent Description Schema.”

Requirements: Maximum 75 words

5 Path Finding Warmup (Total marks: 10)

Imagine a network of cities connected by direct flight routes, where each flight has a cost associated with it. This network is depicted in the graph below, which includes cities labeled with uppercase letters and flight costs labeled on the edges.



Part a) [2 marks]

Perform Breadth-First Search starting from city A. List the order in which cities are visited until city G is first reached. **Assume nodes are expanded in alphabetical order.**

Part b) [2 marks]

Using Depth-First Search starting from city A, list the order in which cities are visited until city G is first reached. **Assume nodes are expanded in alphabetical order.**

Part c) [3 marks]

Starting from H, provide the least expensive routes and their total costs to cities B, D, & F.

Part d) [1 mark]

Using Uniform Cost Search (UCS), find the least expensive route from city A to city G.

Part e) [2 marks]

Using A* search, assume heuristic values $h(n)$ to city G are: A(140), B(120), C(100), D(80), E(60), F(40), G(0), H(160). Find the optimal path from city A to city G.

6 Networks and Pathfinding (Total marks: 50)

In this part of the coursework, you will use graph search algorithms on a **text network** derived from a literary corpus.

You are provided with three files:

- `cw1.py` — where you must implement all required functions.
- `cw1.ipynb` — a Jupyter notebook provided for **testing and demonstrating** your implementations. This file should not be submitted.
- `1984.txt` - the full text of George Orwell's *Nineteen Eighty-Four*.

You will process `1984.txt` into a **graph network of words**, where each unique word is represented as a node and each transition between consecutive words forms a directed edge. This network enables exploration of the novel as a space of linguistic and semantic connections.

6.1 Task 1: Longest Path [5 marks]

`print_long_path` is a method that takes the **two words in the text network that are connected by the longest possible path**, and returns that path as a list of words.

6.2 Task 2: Longest Quote [5 marks]

`print_long_quote` is a method that returns the **longest literal quote** within the text network. That is, the longest contiguous sequence of words that appears exactly as in the original text. The function should return the quote as a list of words.

6.3 Task 3: Most Expensive Path [5 marks]

`print_expensive_path` is a method that takes the **two words in the text network that are connected by the most expensive path**, and returns that path along with its total cost as a tuple.

Use the parameter `distance_mode="inverted"` (remember lab2).

6.4 Task 4: Most Expensive Quote [5 marks]

`print_expensive_quote` is a method that returns the **most expensive literal quote** in the text network. That is, the highest-cost contiguous sequence of words that appears exactly as in the original text. The function should return both the quote and its total cost as a tuple.

Use the parameter `distance_mode="inverted"` (remember lab2).

Evaluation: Tasks 1–4 will be evaluated for correctness and performance. Rankings will be computed across all submissions by path length (Tasks 1–2) and total path cost (Tasks 3–4). Marks will be assigned based on percentile ranking.

6.5 Task 5: Heuristic Search (Total marks: 30)

In this final task, you will design and implement **heuristic search algorithms** (inspired by A*) that traverse the same text network to generate new sequences.

You are encouraged to be creative in your heuristic design. Heuristics may incorporate semantic, syntactic, or statistical principles derived from the text.

Part a) Sentence Completion **[10 marks]**

Implement the function `complete_sentence`, which takes a string of the form `"please believe my eyes <CONTENT>."` (this is just an example; your function should be able to handle ANY string of this form) and replaces the `<CONTENT>` token in a given sentence with a coherent sequence of words found via heuristic search in the text network. Returns the full completed sentence as a list of words.

Part b) Sentence Starting **[10 marks]**

Implement the function `start_sentence`, which takes a string of the form `"two <CONTENT> can ask for a solution."` (this is just an example; your function should be able to handle ANY string of this form) and replaces the `<CONTENT>` token with a coherent sequence of words that could plausibly precede the given phrase.

Part c) Heuristic Description **[10 marks]**

In your `report.pdf`, write a concise technical description (maximum 400 words total) explaining the heuristics you developed for Parts (a) and (b). Each heuristic should be clearly defined and justified, describing how it guides the search process.

Evaluation Criteria:

- Correctness and reproducibility of implemented functions.
- Creativity and soundness of heuristic design.
- Coherence of the generated sequences based on your description of your heuristic.
- Code readability and modularity in `cw1.py`.

Note: Implementations must be in `cw1.py`. The Jupyter notebook is for testing and demonstration only and must NOT be submitted (if you do submit it we won't mark it). You can also submit lab `.py` files if you want us to use your implementations in those files when we evaluate your `cw1.py` code.

Important Note on Path Definitions

No Loops or Repeated Nodes

In this coursework, all graph search algorithms must produce paths that do not contain any loops or repeated nodes.

It is acceptable for your search algorithm to temporarily explore nodes multiple times, but the FINAL path returned by your function must visit each node at most once.

To be super clear:

- Each node may be visited **at most once** in the FINAL path that your functions return.
- Any solution that revisits nodes in the FINAL path it returns (i.e., contains cycles) will be **penalized**.
- You should explicitly prevent loops from existing in the FINAL path found in your search algorithm logic.

The focus of this assignment is on **pathfinding**. Therefore, apply the standard pathfinding principles of no node repetitions and no cycles that we covered in lecture.

This rule applies to all tasks (e.g., *Longest Path*, *Most Expensive Path*, *Quotes*, and *Heuristic Search*).