

Instructions

This assignment is designed to assess your ability to apply the concepts introduced during Weeks 1 and 2 of lectures and tutorials. It consists of three key components: (1) applying the Engineering Method to systematically approach and address the challenge; (2) leveraging appropriate technologies to support your solution; and (3) integrating an AI agent as part of the final implementation. To complete this assignment successfully, you will need to demonstrate your understanding by synthesising and applying what you have learned so far in this unit to design a practical, well-reasoned, and functional solution.

Your Challenge PART 1: On the Engineering Method

Using the engineering method, you will:

1. Analyse the problem and define its requirements.
2. Organise and describe all the data and inputs.
3. Design an algorithm to control the system.
4. Implement the solution using plain English.
5. Test and refine your solution with example input values.

Step-by-Step Instructions

Background:

As part of a transportation infrastructure team, you are tasked with designing a logic-based safety system to control the operation of gates at a railway level crossing. This system must ensure that gates are lowered when a train is approaching or a vehicle is still on the tracks and only raised when it is completely safe.

Your design will rely entirely on simple intuitive logic (Boolean logics, digital logic gates, and truth tables will be taught soon, so don't have to use any of these for this assignment), and will be implemented using an algorithm in plain English.

You are to complete the following steps:

Step 1: Exploring the Problem

1. Restate the problem in your own words.
2. Identify and describe all inputs and outputs of the system.

3. Describe the context, constraints (technical, economic, social, environmental, legal), and stakeholders you can think about.

Step 2: Exploring Alternatives

1. Brainstorm at least two different logic-based solutions.
2. Research at least one real-world example of a crossing gate control system. Please be brief (100 words).

Steps 3 and 4: Evaluating Alternatives + Engineering Decision

- Justify your choice based on simplicity, safety, and reliability.

Step 5 and 6: Planning and Implementing

- Write a sequence of tasks written using plain English.
- Include a flowchart of the logic.

Step 7: Testing and Refinement

- Propose at least four test cases based on input combinations.
- Record expected vs. actual outputs.
- Suggest improvements or refinements to the logic circuit.

Your Challenge PART 2: On the Use of Technology.

To promote professional reporting and personal development practices, you are now required to use [GitHub](#) to manage your project files and collaborate efficiently when needed.

Step 1: Set Up Your Repository

- Create a public or private repository on [GitHub](#).
- Add your tutor and your lecturer as collaborators under *Settings > Collaborators*.
- Use a clear and descriptive repository name, e.g., `pet-feeder-project`.

Step 2: Organize Your Repository:

- Create folders for each step, e.g., `/Step1_Analysis`, `/Step3_Flowchart`, `/Step4_Word_Code`, etc.
- Include a `README.md` file describing your project.
- Use meaningful comments to document changes.

Step 3: Document Your Work:

-

- Push your flowchart (e.g., exported PNG/PDF) to the repository.
- Include your word-based code under a clearly named file.
- Upload test results and any additional documentation.

Submission Requirement:

- Include the GitHub repository link in your report.
- Ensure the repository is accessible to your tutor and unit convener by the due date.

Your Challenge PART 3: On AI Agent Integration

In this part of the assignment, you will explore how Artificial Intelligence (AI) can assist in solving problems, refining logic, and enhancing your assignment's documentation. **You are encouraged to use an AI agent** such as Microsoft Copilot to support your work.

Use Copilot to assist with at **least two** or more of the following:

1. **Refine your logic or Word Code:** Ask Copilot to review your implementation step and suggest improvements or identify potential issues.
2. **Generate alternative solutions:** Prompt Copilot to propose different ways to solve the problem or enhance your flowchart logic.
3. **Explore real-world implementation:** Use Copilot to discuss how your system could be built using actual hardware (e.g., Arduino, Raspberry Pi).
4. **Improve documentation:** Ask Copilot to help you write a professional README.md file or summarize your project for presentation.
5. **Reflect on ethics and limitations:** Use Copilot to explore the ethical implications of using AI in automated pet care or discuss the reliability of AI-generated suggestions.

Deliverable: Write a short reflection (150–250 words) that includes:

- *Prompts and responses (what you asked and what it responded with).*
- *What insights or improvements it helped you achieve?*
- *How it influenced your final solution or understanding of the problem?*
- *You may include screenshots or excerpts of your interaction with Copilot if relevant.*

Submission Requirements and Checklist

Checklist Part 1

- Clearly labelled sections for each engineering method step.
- Research of 100 words (Step 2).
- Flowchart created with Draw.io (Steps 5 & 6).
- Written Sequence of Tasks (Steps 5 & 6).
- Sample test cases and discussion (Step 7).

Checklist Part 2

- GitHub repository link is included.
- Student's GitHub is well structured.

Checklist Part 3

- Reflection of 150-250 words.

Submission instructions

- **This is a SOLO ASSIGNMENT.**
- Allocated marks are shown in the rubric on page 6 of this assignment.
- Create a folder called “uxxxx_Assignment1” and drop all your files in there (problem-solving process answers, flowcharts, Word documents, and the like).
- Compress (zip) ALL your files and folders created above in one single file called uxxxx_Assignment1.zip. Upload this file on Canvas by the due date using the drop box provided in the corresponding assignment.

Appendix

How to Make a Flowchart using Draw.io

1. Open Draw.io:

Go to [Open Draw](#) and click **Start > Device** (or **Google Drive** if you want to save it online).

2. Create a New Diagram

- Choose "**Blank Diagram**"
- Name it something like PetFeeder_Flowchart

3. Use Flowchart Symbols

- Drag and drop standard shapes from the **General** or **Flowchart** section:
 - **Oval** = Start / End
 - **Rectangle** = Process (e.g., "Dispense Food")
 - **Diamond** = Decision (e.g., "Is it 8:00 AM?")
 - **Arrow** = Connect the shapes logically

4. Build the Logic

- Start with "System ON"
- Include key decisions like:
 - "Is it feeding time?"
 - "Is there food in the container?"
 - "Has the pet eaten?"
- Add actions: "Dispense Food", "Wait 10 minutes", "Send Alert", etc.
- Use arrows to guide the logical flow

5. Add Labels and Colours

- Use colours or text formatting to highlight decision points or alerts
- Keep the layout clean and readable

6. Save and Export:

- Go to **File > Export As > PDF or PNG**
- Include the exported image in your assignment submission

Assignment 1 Grading Rubric

Criteria	Excellent (HD)	Good (CR-D)	Satisfactory (P)	Unsatisfactory (N)	Marks
Step 1: Exploring the Problem	Problem clearly restated; all inputs/outputs well identified; multiple constraints and stakeholders discussed.	Problem mostly restated; most inputs/outputs and constraints identified.	Basic problem restatement; limited discussion of constraints and stakeholders.	Incomplete or unclear restatement; little to no identification of system components.	20
Step 2: Exploring Alternatives	Two or more ideas clearly explained; well-researched real-world example within word limit.	Two ideas explained; example provided with minor detail issues.	Only one idea or weak explanation; real-world example lacks clarity.	No alternatives or research included.	15
Steps 3 & 4: Evaluation and Decision	Clear justification with strong reasoning in terms of simplicity, safety, and reliability.	Justification provided with moderate reasoning across some criteria.	Basic reasoning; only one or two evaluation aspects discussed.	No or poor justification of design choice.	20
Steps 5 & 6: Planning and Implementation	Detailed, logical sequence of tasks and accurate, complete flowchart.	Mostly clear sequence and understandable flowchart.	Some task sequence shown; flowchart lacks clarity.	Sequence missing or illogical; no flowchart provided.	20
Step 7: Testing and Refinement	Four+ test cases; all expected vs. actual results compared; meaningful refinements proposed.	Three test cases with some comparison and basic improvement ideas.	Two test cases; minor suggestions or incomplete output comparison.	Fewer than two test cases; no refinements offered.	15

Presentation and Communication	Excellent structure, spelling, diagrams, and clarity throughout.	Mostly clear and well-organized with minor issues.	Understandable but lacks polish or organization.	Disorganized, unclear or poorly presented work.	10
Part 2: Use of GitHub	Repository is complete, organized with folders, README.md, and detailed commit history; tutor and lecturer added.	Repository mostly complete with minor structure issues; README.md present.	Repository present but poorly structured or missing documentation.	Repository missing or inaccessible.	10
Part 3: Use of AI Agent (Copilot)	Reflection includes 2+ examples with prompts/responses and clear insights; demonstrates critical thinking.	Reflection includes 2 examples and shows moderate insight.	Basic reflection with limited engagement or 1 example.	No AI reflection or irrelevant content.	10
TOTAL					120