Instructions

This assignment will test your knowledge and skills in applying the Solving Problem Process taught during weeks 1 and 2 lectures and tutorials and it consists of three parts. The tasks involved are understanding the problem's business rules, tackling the challenge by implementing the problem-solving process, and proposing a suitable solution using nothing else than common sense. As such, the assignment requires you to integrate and synthesise what you have learnt so far in this unit to design and create a proper working solution.

Background: Automated Pet Feeder System

A local animal shelter is looking for a low-cost, programmable automated pet feeder that can:

- Dispense food for cats and dogs at scheduled times.
- Monitor whether food has been consumed or the amount of food that has been consumed.
- Alert staff if there's an issue (e.g., no food dispensed, food not eaten).

They want a solution that could eventually be implemented using low-cost components (like a servo motor and sensors), but your task is to design and simulate the logic and behaviour of the system first.

Your Challenge PART 1: On the Solving Problem Process

Using the Integrated Problem-Solving Process, 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

Step 1: Understand and Define the Problem (Analyse)

- What features must the feeder include?
- What inputs and outputs are needed (e.g., feeding times, sensors)?
- What are possible assumptions or limitations (e.g., limited memory, one type of pet food)?

Deliverable: Clear problem statement, assumptions, inputs/outputs, and a simple sketch or block diagram of the system.

Step 2: Organise and Describe the Data

- List input types (e.g., real-time clock, food level sensor, weight sensor under bowl).
- List expected outputs (e.g., rotate motor, send alert).
- Provide sample values and operational constraints.

Deliverable: Data table with inputs/outputs and operational parameters.

Step 3: Plan the Solution (Design the Algorithm)

- Create decision logic for dispensing food (e.g., "At 8AM and 6PM, rotate servo to dispense food").
- Add logic to detect errors (e.g., "If bowl weight unchanged 10 mins after feeding, send alert").
- You must create a flowchart to represent your algorithm using **Draw.io**. Follow the instructions shown in the appendix of this assignment.

Deliverable: A flowchart representing your automated pet feeder logic (exported from Draw.io and included in your assignment report). You must also include in your submission the actual Draw.io file.

Step 4: Implement the Solution (Word Coding)

- -Translate your logic into a series of consecutive tasks as shown in the lecture and tutorials.
- -Keep it simple, write a modular sequence with meaningful variable names and comments.

Deliverable: Sequence of tasks with suitable explanations.

Step 5: Test and Refine the Solution (Debug and Verify)

Test your logic with some sample scenarios. Have a look at the following examples:

- Pet eats as expected
- Pet does not eat
- Food bin is empty
- Compare output with expectations
- Suggest improvements

Deliverable: Test outputs, discussion of logic, and system refinements.

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 Step 4 implementation 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 problem-solving step.
☐System sketch or diagram (Step 1).
☐Data table (Step 2).
☐Flowchart created with Draw.io (Step 3).
☐Written Sequence of Tasks (Step 4).
☐Sample test cases and discussion (Step 5).

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 7 of this assignment.
- This assignment requires you to include all supporting files as indicated in the checklists above.
- Create a folder called "uxxxxx_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 uxxxxx_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 (DI)	Satisfactory (CR)	Needs Improvement (NN)	Marks
Part 1: Problem-Solving Process					**/50**
Step 1: Problem Analysis	Clear, detailed problem statement with well-defined assumptions and a professional sketch/block diagram.	Mostly clear with minor gaps in assumptions or diagram clarity.	Basic problem statement with limited assumptions or unclear diagram.	Incomplete or unclear analysis.	/10
Step 2: Data Description	Comprehensive input/output table with realistic sample values and constraints.	Mostly complete with minor omissions.	Basic table with limited examples.	Missing or poorly structured data.	/10
Step 3: Algorithm Design (Flowchart)	Accurate, logical flowchart using correct symbols and structure.	Mostly correct with minor logic or formatting issues.	Basic flowchart with some structural flaws.	Incomplete or incorrect flowchart.	/10
Step 4: Word Code Implementation	Clear, modular sequence with meaningful variable names and comments.	Mostly clear with minor issues in structure or naming.	Basic sequence with limited clarity.	Disorganized or unclear code.	/10
Step 5: Testing and Refinement	Thorough testing with insightful discussion and realistic improvements.	Adequate testing with some discussion.	Basic testing with limited reflection.	Minimal or missing testing.	/10
Part 2: GitHub Documentation					**/30**
Repository Setup	Well-organized repo with clear naming and collaborator access.	Mostly organized with minor issues.	Basic setup with limited structure.	Poorly organized or inaccessible repo.	/10

Documentation & Commit History	Clear README, structured folders, and meaningful commit messages.	Mostly complete with minor gaps.	Basic documentation and commits.	Minimal or missing documentation.	/10
File Submission & Structure	All required files correctly uploaded and structured.	Minor issues in file naming or structure.	Basic submission with some missing elements.	Incomplete or disorganized submission.	/10
Part 3: AI Agent Integration					**/20**
Use of AI Agent (Copilot)	Thoughtful, relevant use of AI with clear impact on solution.	Adequate use with some reflection.	Basic interaction with limited insight.	Minimal or irrelevant use of AI.	/10
Reflection Summary	Clear, well-written reflection showing understanding and learning.	Mostly clear with minor gaps.	Basic reflection with limited depth.	Missing or unclear reflection.	/10
Total					/100