

CSE101 Assignment 1

Design an algorithm for an everyday problem

Observe a situation, describe a solution, and test whether it works. No coding or hardware construction is required.

Individual work. Submit one PDF of up to three pages. The deadline and submission destination will be announced.

Your challenge

Choose one small problem from campus or everyday life. Design a clear procedure that could help someone make a decision or carry out a task. Your goal is to show how your solution works, where it can fail, and how you improved it.

Possible topics include a cafeteria queue, unused classroom lighting, study-room reservations, a crowded campus bus, or plant watering. You may choose another manageable problem. Focus on one decision, such as allocating an available room, rather than redesigning an entire university.

1 Describe the problem

Write one concrete observation and explain who is affected. State what a successful result would look like. A short sketch is enough; photographs and interviews are not required. Do not include identifiable people or private information.

2 Define what your system knows

List the inputs, their possible values or units, the outputs, and any information that must be remembered. Explain how each input could be obtained. State your assumptions and limits: a system cannot act on information it does not have.

For example, “the room is empty” is an assumption unless you explain what observation or input establishes it. If a sensor or a person supplies information, consider what happens when that information is missing or incorrect.

3 Write the algorithm in plain language

Use numbered steps with precise actions and conditions. Include at least one decision with different outcomes. Use repetition only when the problem needs it; explain what changes on each cycle and when the process stops or waits for a new event. Do not add a meaningless loop simply to include one.

Avoid instructions such as “manage the queue intelligently.” State the actual rule, including how you handle ties or unavailable information when relevant.

4 Draw a matching flowchart

Translate your numbered algorithm into a readable flowchart. Use the symbols below, connect them with arrows, and label the branches leaving a decision, for example Yes and No. Every outcome should lead somewhere clear.



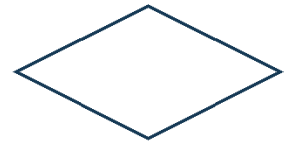
Start or end



Action



Input or output



Decision

A neat hand-drawn flowchart is fully acceptable. A drawing tool is optional. You are assessed on logic and readability, not artistic skill. Keep the diagram consistent with your final numbered steps.

Check every path: can an action be reached, can a decision be answered using available inputs, and does a repeated process have a meaningful stopping or waiting condition?

5 Test your idea on paper

Choose three specific cases. First state what should happen according to your goal, then follow your algorithm step by step. Record the actual outcome of that trace and compare it with the expected outcome. No execution or simulation software is needed.

Case	What to investigate
Ordinary case	A typical situation in which all required information is available.
Boundary case	An exact limit or a change of condition, such as the last available seat, zero requests, or exactly the chosen threshold.
Failure case	Missing or incorrect information, an unavailable resource, or a situation where a simplifying assumption fails.

For each case, include concrete inputs, the expected outcome, the important steps or decision path, and the resulting outcome. A compact table or three short paragraphs is sufficient.

6 Find a weakness and improve the design

Challenge your first version. Identify one specific weakness, explain its consequence, and describe a change that addresses it. Show the final algorithm and flowchart, and briefly explain the earlier rule you replaced. Recheck the affected case after the change.

As a thinking example, “turn off the lights when no movement is detected” may fail when people sit still during an exam. What can the system actually observe, and what extra rule or input would help? Develop and justify your own solution rather than copying this example.

Finish with one limitation that still remains. A clear, honest limitation is more useful than claiming your design handles every possible situation.

What to submit

Prepare one PDF of up to three pages, including diagrams and references. Put your name and student number on the first page. Handwritten work is acceptable if the scan or photograph is legible. No program, electronic circuit, paid tool or working prototype is required.

Suggested organization: page 1 for the problem, inputs and numbered algorithm; page 2 for the flowchart; page 3 for the three test cases, revision and remaining limitation. You may arrange these differently if everything is easy to follow.

Marking rubric

Criterion	Points	What earns credit
Problem and success condition	15	A concrete, manageable problem with a clear goal.
Inputs outputs and assumptions	15	Information is defined and realistically obtainable; limits are stated.
Algorithm	20	Precise steps, meaningful conditions, and sensible handling of stopping or waiting.
Flowchart	20	Readable symbols, labeled branches and complete paths that match the algorithm.
Three paper tests	20	Specific cases with expected outcomes, trace evidence and comparisons.
Revision and reflection	10	A justified improvement, an affected case rechecked, and a remaining limitation.
Total	100	Clarity and reasoning matter more than visual polish.

Before you submit

- My procedure can be followed without guessing what I mean.
- My algorithm and flowchart describe the same final solution.
- I tested an ordinary case, a boundary case and a failure case.
- I explained one improvement and one remaining limitation.
- I identified any sources, tools or assistance used.

Submission and assistance

Deadline and submission destination: CHECK TEAMS PAGE DEADLINE SUBMISSION IS FROM TEAMS NO OTHER OPTION. Follow the published CSE101 AI-use guidelines and any assignment-specific instructions announced by the instructor. Identify any sources, tools or assistance used.

Course resources: <https://www.mehmetgokturk.com/cse101/>

This is CSE101 homework, separate from CSE107 prelabs and lab reports.