SENG 11232 · Engineering Foundation

ASSIGNMENT 01

# Foundations of Computing & Professional Practice

Individual Assignment — Modules 1 & 2 · Lectures 1–6

| | |
|---|---|
| **Course** | SENG 11232  Engineering Foundation |
| **Assignment** | 01 of 02 |
| **Weighting** | 25% of final grade |
| **Submission** | Week 7 — submitted by 23:59 on the Friday of Week 7 (exact date to be confirmed) |
| **Format** | Single PDF document, maximum 3000 words (excluding diagrams, code and tables) |
| **Word Limit** | 3000 words |

## Overview

This assignment covers the first two modules of SENG 11232: Thinking Like an Engineer (Lectures 1–3) and Systems & Digital Foundations (Lectures 4–6). You will demonstrate your understanding of engineering ethics, computational thinking, mathematical foundations, data representation, digital logic and computer architecture.

This is an individual assignment. All tasks must be completed independently. Where code is required, any standard programming language may be used; pseudocode is acceptable unless a specific language is stated.

Submit your answers as a single PDF document unless otherwise specified.

## Tasks

### Task 1: Engineering Ethics Case Analysis [25 marks]

Read the following scenario carefully, then answer the questions below. Scenario: A software team at a financial services company discovers, one week before a major product launch, that their trading algorithm has a rare timing bug that could cause incorrect trades under specific high-load conditions. The probability of the bug manifesting is estimated at 1 in 50,000 transactions. Management decides to proceed with the launch, arguing the risk is acceptable and the fix can be deployed in a post-launch patch. You are the lead software engineer on the team.

**Your submission must include:**

- Part A (10 marks): Identify which principles of the ACM/IEEE Software Engineering Code of Ethics are most relevant to this scenario. For each principle you identify, explain how it applies and whether you believe it is being upheld or violated in management's decision.
- Part B (8 marks): As the lead engineer, what actions would you take? Provide a structured argument for your position, including: (i) technical steps, (ii) communication to management, and (iii) any external obligations you may have.

- Part C (7 marks): Compare this scenario to one of the real-world failures covered in Lecture 1 (Therac-25, Ariane 5, Boeing 737 MAX, or Post Office Horizon). Identify at least two parallels and one significant difference.

| GUIDANCE | Strong answers will engage with the specific technical risk (not just generic principles), reference the Code of Ethics precisely, and demonstrate moral courage — the willingness to act even under pressure. |
|---|---|

### Task 2: Computational Thinking & Algorithm Design [30 marks]

This task tests your ability to apply computational thinking to a non-trivial problem.

**Your submission must include:**

- Part A — Decomposition (10 marks): A university wants to build a digital lost-property system. Students and staff can report lost items, search for found items, claim items and receive notifications when a match is found. Apply top-down decomposition to this system. Draw a hierarchical diagram showing at least 3 levels of decomposition. Identify 4 data entities and their key attributes.
- Part B — Algorithm Design (12 marks): Design an algorithm (in pseudocode) that searches the lost-property database for potential matches between a reported lost item and all found items. Your algorithm must: accept a lost item description as input; compare it against all found items using at least two matching criteria (e.g. category, date, location); return a ranked list of potential matches; handle the case where no matches exist. Include an analysis of the time complexity of your algorithm (informal Big-O is acceptable).
- Part C — Mathematical Foundations (8 marks): Express the following as formal predicates: (i) Every lost item in the system has exactly one owner; (ii) There exists at least one found item with no matching lost report; (iii) If an item is claimed, it has a matching lost report. Using set theory, describe the relationship between the set of lost items and the set of found items in terms of union, intersection and difference — and explain what each subset means in the real-world context.

| GUIDANCE | Pseudocode does not need to follow a specific language syntax, but must be unambiguous enough that another person could implement it. For Part C, use formal notation (∀, ∃, ∈, ∪, ∩, \). |
|---|---|

### Task 3: Data Representation & Digital Logic [25 marks]

Demonstrate your understanding of data representation and digital logic through calculation and analysis.

**Your submission must include:**

- Part A — Number Systems (10 marks): (i) Convert 237 decimal to 8-bit binary. Show your working. (ii) Convert 0xB4 to decimal and to octal. (iii) Represent -95 in 8-bit two's complement. Verify by adding -95 and +95 in binary and confirming the result is zero. (iv) A floating-point addition produces 0.1 + 0.2 = 0.30000000000000004. Explain why this occurs and state two approaches a software engineer should use to handle monetary calculations correctly.
- Part B — Boolean Algebra & K-Maps (15 marks): A traffic light controller has inputs: E (emergency vehicle detected, 1=yes), N (night mode, 1=night), P (pedestrian button pressed, 1=pressed). The output G (green light for main road) should be ON when: emergency vehicle detected regardless of other inputs; OR normal day conditions with no pedestrian request. (i) Write the complete truth table for G(E, N, P). (ii) Write the Boolean expression from the truth table in Sum of Products (SOP) form. (iii) Use a Karnaugh map

to minimise the expression. (iv) Draw the minimised gate circuit.

### Task 4: Computer Architecture Short Questions [20 marks]

Answer each of the following short questions clearly and concisely.

**Your submission must include:**

- (a) [5 marks] Trace the fetch-decode-execute cycle for the instruction: LOAD R1, [200] (load the value at memory address 200 into register R1). Describe each step, identifying which CPU component is involved.
- (b) [5 marks] A developer writes a function that accesses elements of a large array in random order rather than sequential order. Explain why this might be significantly slower than sequential access and what hardware mechanism is responsible.
- (c) [5 marks] Explain the difference between a compiled language (e.g. C) and a JIT-compiled language (e.g. Java) in terms of how source code becomes executable. What is one advantage of each approach for a software engineer?
- (d) [5 marks] Explain what a system call is, give two examples of operations that require system calls, and explain why user programs cannot access hardware resources directly.

## Submission Instructions

- Submit via the course Learning Management System (LMS) assignment portal
- File naming: SENG11232_A1_[YourStudentID].pdf
- All diagrams must be clearly labelled with titles and legends
- Mathematical notation must be clear and consistently used
- Pseudocode must be indented consistently to show block structure
- Late submission policy: 10% deduction per 24-hour period after the deadline
- Requests for extensions must be submitted at least 48 hours before the deadline

**NOTE** Academic integrity: all submitted work must be your own. Use of AI tools to generate your submitted answers without acknowledgement constitutes academic misconduct. You may use AI as a thinking tool (to get feedback, generate counter-arguments), but the analysis, reasoning and writing must be yours.

## Marking Scheme

| Task | Description | Marks |
|---|---|---|
| Task 1A | Code of Ethics principles identified and applied | 10 |
| Task 1B | Engineering response — technical and professional | 8 |
| Task 1C | Comparison with real-world failure | 7 |
| Task 2A | Decomposition diagram and data entities | 10 |
| Task 2B | Algorithm design in pseudocode with complexity | 12 |
| Task 2C | Formal predicates and set theory application | 8 |
| Task 3A | Number system conversions and floating-point analysis | 10 |
| Task 3B | Truth table, Boolean simplification and K-map | 15 |

| | | |
|---|---|---|
| Task 4 | Architecture short questions (a)-(d) | 20 |
| | **TOTAL** | **100** |