INF1009 Final Report and Presentation Templates

1. Project Report (10%)

Your report should clearly describe **how your Abstract Engine program works, how it was developed, and how your team contributed**.
It should be **concise, well-structured, and professional**, showing clear understanding of your design, logic, and testing.

Length: 10-15 pages (single column, font size 11 or greater, standard margins).
The page count excludes the title page, table of contents, and appendix.
The report must be **self-contained** - if the examiner needs to look into the appendix to understand your work, that content will count toward the main page limit.

The structure below is to help you structure your report; it is not an exact template.

Recommended Structure

1. Title Page

Include:

- Project title
- Team ID and team name
- Team members (name, student ID, email)
- Date of submission

2. Introduction (1 to 2 pages)

Briefly introduce the system you built:

2.1 Overview of the Abstract Engine

- Purpose of the engine
- What the engine aims to support
- Overall vision

2.2 Key Features Implemented

- Must-have features
- Any unique innovations or enhancements
- High-level summary of systems or components

3. System Design and Implementation (6–8 pages)

Describe **how your system is structured** and **how it works internally**.

Include:

3.1 Architectural Overview

Describe your chosen design style:

- Class-Based Approach *(explain rationale)* OR Component-Based Approach *(explain rationale)*
- High-level diagram or flow diagram
- Major subsystems and their interactions

3.2 Managers and Their Responsibilities

For each manager:

- Name of Manager
- Purpose
- Objects Managed
- High-Level Methods Only

3.3 Application of OOP Principles

Explain how you applied:

- Abstraction
- Encapsulation
- Inheritance (IS-A)
- Composition (HAS-A)
- Interfaces / Abstract Classes
- Polymorphism (Override/Overload)

4 UML Diagram (Full Engine UML)

Include a complete UML covering:

- Entities / Components
- Managers
- Scene / Core System
- Relationships (IS-A, HAS-A)
- Dependencies
- Interfaces / abstract classes

5. Team Contributions (1 page)

Summarize what each team member worked on. Remember, here you must also reflect on contributions from your AI-Teammate

For example:

Team Member	Major Contributions
A	UML design, Scene Manager, Rendering flow
B	Entity system, OOP review
C	Input/Output Manager, Integration
D	Documentation, report integration, debugging
AI	Architectural suggestions, clarification support, ideation for unique features

---

Presentation and Demonstration Recommended Structure (10%)

- Duration: 8–10 minutes (strict)
- Slides: 10–12 slides
- Video must be clear, well-paced, and well-structured
- The presentation must be self-contained
- Anyone watching the video should fully understand your engine without needing to read the report.
- Demonstration of the Abstract Engine is mandatory

Your team must submit a **video presentation** that includes:

Participation / Presentation

- Every member must speak briefly to practice presentation skills.
  You may divide sections as you prefer (e.g., each person covers 1–2 functions).

Architecture Overview and Key Parts

- Explain how the engine is structured
- Focus on the **logic flow** — don't explain every line.
- Briefly describe managers and entities/components.
- Highlight how OOP principles are applied (inheritance, composition, abstraction).

UML Diagram

Point out the main relationships and design decisions.

Abstract Engine Demonstration

- Show your Abstract Engine running live.
- The engine initialization
- Scene creation
- Manager coordination
- A simple simulation loop
- Evidence that the engine runs with no errors
- Demonstrate all required functions, enhancement features and unique feature.

Time Management

- Keep it concise but informative — Max of **10 minutes total**.
- Ensure transitions between presenters are smooth and clear.