

Title of Module	ELEE1147 Programming for Engineers
Title(s) of Assignment	Coursework
Module Team	Seb Blair
Assignment Deadline	7 <sup>th</sup> April 2025 at 23:30
Assignment Weighting	70%
Expected Feedback Date	22 <sup>th</sup> April 2025

## AI Guidance and Plagiarism

All submitted work will be electronically scanned for plagiarism and the use of Artificial Intelligence (AI) software. The work that you submit must be your own; any material from other sources must be correctly cited and referenced. If unreferenced material is detected a student will be reported for plagiarism. AI use is discouraged in the production of this work. Use of AI which has not been acknowledged is an academic misconduct offence and again will lead to a report of plagiarism. University regulations require that this fact is clearly stated in the Report

## Overview

The purpose of this assignment is to apply your knowledge of C, Python, and version control systems to identify, fix, and enhance an existing codebase. You will work with a provided codebase that contains intentional issues (e.g., compilation errors, runtime errors, and logic flaws) and enhance it by applying modern software engineering practices. Your task is to document your problem-solving process, demonstrate proficiency in using Git, and adhere to industry best practices.

You are provided with a legacy codebase containing both C and Python components that perform a specific operation (e.g., data processing, mathematical computation, or system control). The codebase is functional but intentionally contains a variety of errors and lacks modern practices. Your task is to:

- **Identify and fix the existing errors** (compilation, runtime, logical) in the codebase.
- **Refactor and enhance** the code to improve its performance, maintainability, and scalability.
- **Document your process** for fixing the errors and enhancing the code.

## Assignment Instructions

### Tasks

#### 1 Initial Setup and Version Control:

- **Clone the Repository:** Start by cloning the provided Git repository that contains the faulty C and Python codebase.

\* <https://classroom.github.com/a/eFDQuzIZ>

**2 Identify, Log, and Categorize Errors:**

- Compile-Time Errors
- Runtime Errors
- Logical Errors

**3 Refactoring and Enhancement:**

- Refactor the Code
- Enhance Functionality
- Commenting the Code

**4 Version Control and Branch Management:**

- Incremental Branching
- Commit Regularly
- Merging Branches

**5 Documentation, Error Logs, and Best Practices:**

- Document Your Process
- Log Your Fixes
- Adhere to Best Practice
- Provide Evidence

**6 Report Writing:**

- Understanding the Code
- Refactoring and Enhancement Process
- Challenges and Solutions
- Final Summary

## Submission Guidelines

Submission link:

- <https://moodlecurrent.gre.ac.uk/mod/turnitintooltwo/view.php?id=2869466>

Your submission should be made up of these two separate parts and submitted to the respective submissions points:

- 1 Final Repository:** The commit closest to the submission date and time will be reviewed. Ensure that the repository is well-organized and that the main branch reflects the completed project.
- 2 Report:** Submit a PDF report summarising your work on Moodle, including the detailed documentation, categorised error log, and a reflection on the process.

## Mark Scheme/Rubric

Please see page 4 for a detailed rubric.

Marks %	— Overview of the rubric.
0 - 30	<a href="#">Error Identification and Resolution</a>
0 - 20	<a href="#">Version Control and Branch Management</a>
0 - 25	<a href="#">Code Refactoring and Enhancement</a>
0 - 15	<a href="#">Documentation and Logging</a>
0 - 10	<a href="#">Report Writing</a>

Table 1: Overview of the rubric.

## APPENDIX A: MARK RUBRIC

Marks (%)	Rubric for Error Identification and Resolution (30%)
0 - 29	Errors are not identified, or corrections are ineffective. Minimal or no attempt made.
30 - 39	Major errors identified but not effectively resolved. Significant gaps in understanding.
40 - 49	Basic errors identified and resolved, but with several mistakes or incomplete solutions
50 - 59	Most errors identified and resolved with minor mistakes. Shows adequate understanding
60 - 69	Good identification and resolution of errors with few mistakes. Demonstrates good understanding.
70 - 79	Almost all errors are identified and resolved effectively. Minor mistakes, strong understanding shown.
80 - 100	All errors are identified and resolved with precision. The student demonstrates deep understanding and excellent problem-solving skills.

Table 2: Rubric for: Error Identification and Resolution (30%).

Marks	Rubric for Version Control and Branch Management (20%)
0 - 29	No meaningful use of version control. Little or no branching strategy .
30 - 39	Basic use of version control with significant mistakes. Poor branching strategy.
40 - 49	Version control used with some errors, basic branching strategy applied.
50 - 59	Good use of version control with minor mistakes. Branching strategy is somewhat clear.
60 - 69	Very good use of version control, clear branching strategy with few issues.
70 - 79	Excellent use of version control, with a well-thought-out branching strategy. Minimal issues.
80 - 100	Exemplary use of version control, including a sophisticated branching strategy. No noticeable issues.

Table 3: Rubric for Version Control and Branch Management (20%).

Marks	Rubric for Code Refactoring and Enhancement (25%)
0 - 29	No refactoring or enhancement made, or the changes degrade the code.
30 - 39	Minimal refactoring with significant issues. Enhancements, if any, are poorly executed.
40 - 49	Basic refactoring and enhancements made with some mistakes or minimal impact.
50 - 59	Adequate refactoring and enhancement, showing good improvement in code quality.
60 - 69	Good refactoring with clear enhancements. Code is more readable and maintainable.
70 - 79	Very good refactoring and enhancement, with significant improvements in code quality.
80 - 100	Exceptional refactoring and enhancement, resulting in highly efficient, readable, and maintainable code.

Table 4: Rubric for Code Refactoring and Enhancement (25%)

Marks	Rubric for Documentation and Logging (15%)
0 - 29	No documentation or logging provided, or the documentation is irrelevant.
30 - 39	Poorly documented, with minimal or unclear logs.
40 - 49	Basic documentation and logs provided, but with significant gaps or unclear explanations.
50 - 59	Adequate documentation and logs, with some minor gaps or unclear sections.
60 - 69	Good documentation and logging, with clear explanations and few gaps.
70 - 79	Very well-documented and logged, with clear, thorough explanations and minimal gaps.
80 - 100	Exemplary documentation and logging, comprehensive, clear, and insightful.

Table 5: Rubric for Documentation and Logging (15%)

Marks	Rubric for Report Writing (10%)
0 - 29	No report provided, or report is irrelevant and lacks clarity.
30 - 39	Report is incomplete, with poor clarity and little relevance to the task.
40 - 49	Basic report provided, but with significant gaps in analysis and clarity.
50 - 59	Adequate report, with some analysis and clarity, but with minor issues.
60 - 69	Good report, with clear analysis and well-organized content. Few minor issues..
70 - 79	Very good report, with thorough analysis, clear structure, and minimal issues.
80 - 100	Exceptional report, with deep analysis, excellent clarity, and a well-organized structure.

Table 6: Rubric for Report Writing (10%)