National University of Sciences & Technology (NUST)
School of Electrical Engineering and Computer Science (SEECS)
Faculty of Computing

CS 110: Fundamentals of Computer Programming
Semester Project

Introduction:

For your Fundamentals of Programming capstone project, you and your team will design and develop a functional software application that creatively solves a specific problem or provides a meaningful experience for users. This project will challenge you to apply programming skills you've learned, use logical structures effectively, and collaborate with your team to create a unique solution. Your application should be developed in C++ within Visual Studio IDE, leveraging structures for data organization and manipulation. The scope of the project is intentionally open-ended, encouraging you to approach your chosen problem creatively and think beyond standard solutions.

Project Goals:

- **Problem-Solving**: Identify a real-world problem or an engaging user need and create an application that provides a thoughtful and functional solution.
- **Creativity & Uniqueness**: Aim to develop an application that stands out in its functionality, user interaction, or unique approach to problem-solving. Projects should be sufficiently open-ended to avoid overlaps between groups.
- **Team Collaboration**: Work collaboratively with your group members, each contributing to different aspects of the project—such as logic, user interaction, and data handling—to achieve a cohesive and well-rounded application.
- **User Experience**: Focus on creating an application that is user-friendly and meets its intended purpose effectively, with clear prompts and logical user flows.

Project Ideas:

Choose from one of the following project areas as a starting point or propose your own idea within the same scope. Each option includes a basic outline but leaves space for your team to explore, add unique features, and deepen the project's complexity.

- **Financial Budget Tracker:** Design an application to help users manage and categorize personal finances. Track income, expenses, and generate budget summaries. **Additional Ideas**: Add budgeting goals, notifications, and customizable categories.
- **Customizable Quiz Application:** Develop a dynamic quiz tool where users can create and take quizzes on any topic. **Additional Ideas**: Allow users to save quizzes, track scores, or share quizzes with others.

- **Emergency Resource Locator:** Build an app that helps users locate nearby resources in emergencies. Store data on hospitals, shelters, and other resources. **Additional Ideas**: Add a location tracker, or create a "status update" feature for resource availability.
- **Exercise and Health Tracker:** Create an application to track workouts, health metrics, and personal goals. **Additional Ideas**: Add personalized feedback based on activity, visual progress charts, or daily reminders.
- **Eco-Friendly Product Recommender:** Design a tool that recommends sustainable alternatives to common products. **Additional Ideas**: Let users contribute their own suggestions or add a scoring system for environmental impact.
- **Traffic Light Simulation:** Develop a simulation of a small traffic system to manage intersections. **Additional Ideas**: Implement varied traffic patterns or add congestion reports for different times of day.
- **Household Expense Splitter:** Build an app that helps roommates track and split shared expenses. **Additional Ideas**: Enable uneven splits, show running balances, or allow expense categories for better tracking.

Core Requirements:

- **Structured Data Management:** Use structures to store and manage data (e.g., items in an inventory, quiz questions, or traffic data). Ensure data is well-organized, easy to access, and supports all necessary operations (add, update, delete, search).
- **User Interaction and Navigation:** Implement a simple and intuitive text-based user interface. Provide clear prompts and feedback, guiding the user through the application flow (e.g., moving through budget categories, quiz options, health tracker entries). Include instructions to make the interface user-friendly.
- **Progress Tracking and Reporting:** Track and display progress or achievements (e.g., exercise goals met, quiz scores, expenses saved). This can include visual elements like progress bars, summaries, or statistics to show the user's accomplishments.
- **Customization Options:** Allow users to tailor some aspects of the application to their needs (e.g., adding new categories in budget trackers, creating custom quiz topics, setting specific health goals, or personalizing eco-friendly product recommendations).
- **Randomized Elements:** Incorporate elements of randomness to enhance engagement, such as:
    - Randomized challenges in quizzes.
    - Random events in health tracking, like surprise fitness challenges.
    - Changing traffic patterns in the simulation.
    - Decision Points and Outcomes
    - Enable user choices that influence the application’s behavior or outcomes:
    - Budget decisions that affect savings tracking.
    - Multiple answer choices in quizzes that influence scoring and feedback.
    - Health choices that affect weekly summaries or overall health rating.

- **Data Persistence:** Allow users to save and load data (e.g., budget history, quiz questions, traffic patterns). Use basic file handling in C++ to persist data between sessions, so users can resume their progress.
- **Performance Metrics and Scoring:** Implement a scoring or evaluation system to assess user performance:
- **Expense tracking summary (e.g., percentage of budget spent):** Quiz scores or health goals achieved. Traffic efficiency or congestion tracking in simulations.
- **Feedback and Error Handling:** Provide real-time feedback on user actions, including validation for inputs (e.g., handling invalid entries for expenses, non-numeric quiz answers). This improves usability and prevents errors.
- **Progressive Difficulty or Levels:** Implement progressive difficulty or advanced options such as follows:
    1. Higher difficulty quiz questions or budgeting challenges.
    2. Increasing traffic complexity in simulations.
    3. Progressive health goals or exercise challenges.
- **Summary and Reporting:** At the end of each session, provide a summary report that includes key insights or metrics:
    - Budget summaries with expenses by category.
    - Quiz results with correct/incorrect answers.
    - Weekly health reports with total calories burned, water intake, etc.
- **Optional: Visual Components:** Although a full graphical interface is not required, you may include simple text-based visual aids or ASCII art to enhance the user experience, like progress bars, icons, or dividers.

Project Deliverables:

The following deliverables are required for the project.

Phase I: Project Proposal (Deadline: 9th December 2025):

In the first phase, you are required to submit your project proposal in the form of presentation (PPT) that outlines your team's concept, problem statement, objectives, and initial research. Present your ideas in the lab in the form of a presentation. The content of the presentation must include following (but are not limited to):

- Problem statement, proposed solution, key features, and tentative technical requirements.
- Clearly defined scope of the project and milestones for development.

Once ideas are finalized, you will proceed to the design phase**.**

Phase II: Technical Design Report (Deadline: 16th December 2025):

In this phase, you will be required to submit the detailed design of the solution you are developing. You need to present your solution design as a technical design report. The content of the document/report must include the following (but are not limited to):

- High-level design diagram or flow chart showing complete flow of processes from input to output.
- Algorithmic/low-level design that explains each module or function.
- Tentative user interface, use-case diagrams to explain how user will interact with the software.
- Functional and non-functional requirements.
- Tools and techniques to be employed.

Phase III: Implementation (Deadline: 23rd December 2025):

In the final phase, you will be required to submit the following:

- **Source Code:**

  Develop the application using the programming constructs covered in the semester (You can use additional with justification, however, do not use classes.)

  Include comments within the code for clarity.

- **User Manual:**

  Create a user manual explaining how to use the software. This can be a Readme file included in your Git repo.

- **Presentation/Demo:**

  Present the project to the class, demonstrating key features and functionalities. Discuss design decisions, challenges faced, and lessons learned during development.

- **Final Report:**

  Submit a final report summarizing the project's overall process from idea inception and design till development. Test your software against various test-cases and show outcome. Also summarize success, challenges, and future improvements. Reflect on the learning outcomes achieved through the project.

Learning Outcomes and Objectives:

The following learning outcomes and objectives are targeted in this semester project.

CLOs	**Description**
CLO 2	**Solve** given real-world problem by applying appropriate programming concepts and techniques
CLO 3	**Build** a program and associated documentation using appropriate IDE and supplementary tools
CLO 4	**Perform** effectively as an individual and as a member of a team.

The project will enable students to build a comprehensive solution to a real-world problem in C++ using Visual Studio or VS-Code IDEs. The students will explore various libraries to provide an interactive experience to the users. In addition, students will be creating a Readme file or user manual for the users to understand the technical requirements to execute the software on their systems as well as instructions. The software design and implementation will be documented to support developers in recreating the application. The students' ability to use the IDE and to create supporting documentation will be assessed by the final demonstration and report.

- **Activities Mapped on CLO-2:** Technical Design Report and Code
- **Activities Mapped on CLO-3:** Final Demo and Project Report

This project is a group activity and requires students to work in a team. This will give them a chance to collaborate with their peers. Through the different stages of the project starting from proposal to final deliverables, students will be assessed on their teamwork, balanced task division, willingness to resolve disagreements, and ability to lead and follow instructions of the team leader.

- **Activities Mapped on CLO-4:** Presentations of Phase-I and Phase-III

Projects will be evaluated on the following criteria:

- Proposal Presentation (CLO-4) 20%
- Design Report (CLO-2) 20%
- Final Presentation (CLO-4) 20%
- Project Demo (CLO-3) 20%
- Project Report (CLO-3) 20%

Submission Criteria:

This is a group assignment. Both members should submit the same deliverables in LMS no later than the deadlines mentioned above. **Late submissions are not accepted. Plagiarism will be marked zero. Give credit where due. Rubrics for each deliverable will be provided later.**