CS112 Object Oriented Programming Project – Spring 2025

Smart Eco City Simulation Platform

Mapped on CLO1 and CLO2

The project aims to develop a **Smart Eco City Simulation Platform** that provides users with the experience of creating, managing, and improving an environmentally sustainable virtual city. This simulation will incorporate **gaming elements**, **social engagement**, and **educational content**, while reinforcing key **object-oriented programming** concepts. The platform encourages users to make eco-friendly decisions, simulate city growth, and learn about sustainability in a fun and engaging way.

Project Requirements:

1. **Implement Classes, Constructors, and Inheritance** to model entities such as buildings, citizens, utilities, transportation, resources, and pollution levels.
2. **Utilize Polymorphism** to manage different types of buildings (residential, commercial, green spaces), transportation (bikes, buses, EVs), and activities (recycling, tree planting, energy production).
3. **Implement Virtual Functions** to enable polymorphic behavior in actions such as upgrading infrastructure, responding to environmental alerts, or simulating citizen behavior.
4. **Utilize Class Templates** to manage collections of data such as resource logs, activity histories, and city zones.
5. **Implement Exception Handling** to manage errors gracefully and ensure robustness, such as handling invalid upgrades, budget constraints, or simulation failures.

Project Tasks:

1. **Design the Class Structure**: Define classes for core entities like buildings, transport systems, citizens, power sources, activities, and city zones. Use constructors and inheritance to show relationships (e.g., Residential → Building → CityObject).
2. **Implement Polymorphism**: Use polymorphism to enable different environmental behaviors for various objects (e.g., solar vs. coal plants, EVs vs. diesel buses).
3. **Incorporate Virtual Functions**: Use virtual functions to manage shared yet distinct behavior (e.g., `operate()` for various building types, `move()` for different vehicles).
4. **Utilize Class Templates**: Create generic containers (e.g., `CityLog<T>`, `EcoScore<T>`) to manage dynamic data for various simulations.
5. **Implement Exception Handling**: Add error-handling mechanisms to cover invalid input (e.g., wrong upgrade path, exceeding budget), and simulation anomalies.

6. **Develop User Interface**: (Optional - only if GUI is included in course) Design an interface to manage city settings, visualize eco-score, and display simulation reports.
7. **Gaming Elements**: Introduce eco-points, green levels, rewards for hitting sustainability goals (like zero waste or 100% renewable energy).
8. **Social Interaction**: Allow players to share their cities, compete on eco-rankings, and collaborate in virtual climate summits.
9. **Educational Content**: Embed tips and facts on sustainable living, clean energy, and the impacts of climate change into the gameplay.

Project Evaluation:

- Projects will be evaluated based on adherence to the above requirements, code quality, and creativity in building a functional Smart Eco City Simulator.
- Demos and final presentations will be scheduled during the final week of the term. Each group will demonstrate their work and explain key components of their implementation.
- **Individual evaluation** will consider participation, contribution to the codebase, and understanding shown during the presentation.
- **10% of the final grade** is allocated to this project.
- You may use **AI tools** for help, but **projects generated directly by ChatGPT or similar tools will receive **zero marks**. Build your own logic and apply what you've learned.

Project Deadline:

May 10th, 2025

Group Composition:

- Groups will consist of **5 members** each. Collaborate effectively to design, implement, and present your simulation platform.

Total Marks: 20