

Project: Zoo Simulation System

In this project, you will design and implement an object-oriented simulation of a zoo. The goal is to apply what you have learned about object-oriented programming by building a system where animals and visitors interact over time.

The project is designed with a low entry threshold and a high potential ceiling. Everyone is expected to complete the minimum requirements, while more advanced solutions will naturally extend the system with better structure, robustness, and realism.

Project Requirements

Minimum Requirements (Core Functionality)

- Implement an object-oriented zoo simulation system.
- Include a base Animal class with common attributes such as name, age, and energy level.
- Include subclasses for different animal types (for example Herbivores and Carnivores).
- Implement at least three concrete animal species with unique behaviors.
- Include a Visitor class that can interact with animals (for example feeding them).
- Simulate at least one full day in the zoo where animals eat, sleep, and interact.
- Demonstrate inheritance and method overriding.
- The program must run without errors and clearly show the simulation flow through output.

Extended Requirements (Improved Design & Behavior)

- Manage animal energy levels consistently across all actions.
- Implement meaningful interactions between animals (for example hunting, playing, or resting).
- Ensure that behaviors affect future simulation steps, not just printed output.
- Structure the simulation so that it can run for multiple days without code changes.
- Reduce duplicated logic through proper class design.
- Demonstrate thoughtful use of encapsulation and reuse.

Advanced Requirements (High-Quality Simulation Design)

- Make it easy to add new animal species without modifying existing logic.
- Maintain a balanced predator-prey system that avoids unrealistic extinction.
- Separate simulation logic from execution logic.
- Handle edge cases gracefully, regardless of interaction order.
- Write clean, readable code with clear structure and comments where needed.
- Demonstrate deliberate and well-reasoned design choices.

Optional Extension Tasks

- Allow the simulation to run for a configurable number of days.
- Introduce random events that influence animal or visitor behavior.

- Add a simple text-based interface for user interaction.
- Track and display statistics such as energy levels or population changes over time.

Daily Progress & Communication

You are still expected to submit screenshots showing your progress every day at 16:30.

Do not hesitate to contact us teachers if you have questions, uncertainties, or need guidance at any point during the project. Happy coding 😊