

Object-Oriented Programming

Semester 2025-II

Workshop No. 3 — Applying SOLID Principles to a Domotic Circuit Simulator

Eng. Carlos Andrés Sierra, M.Sc.

Lecturer

Computer Engineering

School of Engineering

Universidad Nacional de Colombia

Building on **Workshop #1** (Conceptual Design) and **Workshop #2** (Technical Implementation), this session focuses on incorporating the **SOLID principles** into your domotic circuit simulator project. You will refine your class designs and implementation plan to ensure your Python code is robust, maintainable, and extensible, following best practices in object-oriented programming.

Workshop Scope and Objectives:

- **SOLID Principles in Practice:** Analyze your current design and demonstrate how each SOLID principle (Single Responsibility, Open/Closed, Liskov Substitution, Interface Segregation, Dependency Inversion) can be applied to your domotic circuit simulator.
- **Refined UML and Design Artifacts:** Update your UML class diagrams and CRC cards to reflect SOLID-driven improvements.
- **Python OOP Implementation:** Provide code snippets in Python that illustrate the application of SOLID principles in your classes and interactions.
- **Critical Reflection:** Explain the benefits and challenges of applying SOLID in the context of a second-semester engineering project.

Carlos Andrés Sierra, Computer Engineer, M.Sc. in Computer Engineering, Lecturer at Universidad Nacional de Colombia.

Any comment or concern regarding this workshop can be sent to Carlos A. Sierra at: *casier-rav@unal.edu.co*.

Methodology and Deliverables:**1. SOLID Principles Analysis:**

- For each SOLID principle, briefly describe its meaning and relevance.
- Identify at least one example in your simulator design where the principle can be applied or improved.

2. Updated UML and CRC Cards:

- Revise your class diagrams and CRC cards to show how SOLID principles have influenced your design.
- Highlight new interfaces, abstract classes, or refactored responsibilities.

3. Python Code Snippets:

- Provide short Python code examples (class definitions, method overrides, interface usage, etc.) that demonstrate your application of SOLID.
- Clearly comment each snippet to explain which principle is being illustrated.

4. Reflection:

- Write a brief reflection (half a page) on the process of applying SOLID principles as a second-semester engineering student.
- Discuss any difficulties, trade-offs, or insights gained.

5. Submission Format:

- Compile your analysis, diagrams, CRC cards, code snippets, and reflection into a single PDF.
- Place all materials in a folder named `Workshop-3` in your course repository, with a `README.md` referencing each section.

Deadline: Friday, November 18st, 2025, at 8:00 PM. Late submissions may affect your grading according to course policies.

Notes:

- All deliverables must be in **English**.
- Cite any references (articles, tutorials) that influenced your SOLID-driven redesign.
- Focus on clarity and the practical application of SOLID to ensure a flexible, maintainable codebase as your domotic circuit simulator evolves.
- Use Python and OOP best practices suitable for second-semester engineering students.

This workshop will help you build a strong foundation in software design, preparing you for more complex projects in your engineering career. Good luck!