

COURSE DESCRIPTION

Object-Oriented Programming

Author: Eng. Carlos Andrés Sierra, M.Sc.
cavirguezs@udistrital.edu.co

Lecturer
Computer Engineering
School of Engineering
Universidad Distrital Francisco José de Caldas

2025-I



Outline

- 1 You don't know who I am
- 2 Course Overview
- 3 Syllabus
- 4 Grading & Rules
- 5 Bibliography



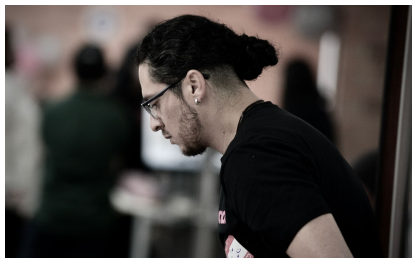
Outline

- 1 You don't know who I am
- 2 Course Overview
- 3 Syllabus
- 4 Grading & Rules
- 5 Bibliography



Academic Experience

- **Computer Engineer, MSc.** in Computer Engineering, and researcher for the last 15 years.
- 7 years as full-time professor at colleges, for Computer Engineering programs.
- 3 years as lecturer professor for both colleges and government STEM programs.
- Speaker in Colombia, Brasil, Bolivia, at IEEE events and colleges.



Academic Experience

- **Computer Engineer, MSc.** in Computer Engineering, and *researcher* for the last **15 years**.
- 7 years as **full-time professor** at colleges, for **Computer Engineering programs**.
- 3 years as **lecturer professor** for both colleges and **government STEM programs**.
- **Speaker** in Colombia, Brasil, Bolivia, at **IEEE** events and colleges.



Academic Experience

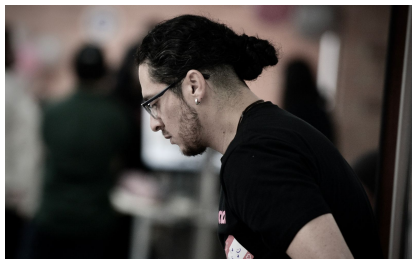
- **Computer Engineer, MSc.** in Computer Engineering, and *researcher* for the last **15 years**.
- 7 years as **full-time professor** at colleges, for **Computer Engineering programs**.
- **3 years** as **lecturer professor** for both colleges and **government STEM programs**.
- **Speaker** in Colombia, Brasil, Bolivia, at **IEEE** events and colleges.

MISIONTIC



Academic Experience

- **Computer Engineer, MSc.** in Computer Engineering, and *researcher* for the last **15 years**.
- 7 years as **full-time professor** at colleges, for **Computer Engineering programs**.
- 3 years as **lecturer professor** for both colleges and **government STEM programs**.
- **Speaker** in Colombia, Brasil, Bolivia, at **IEEE** events and colleges.



Non-academic Experience



- PyCon Colombia and Python Bogotá **co-organizer**.
Collaborations in ScipyLATAM and Jupyter LATAM.
- +3 years performed as **Software Engineer** for several **tech companies** in Colombia.
- 3 years performed as **Technical Leader** of **Machine Learning** and **Data Science** in an **USA startup**.
- +1 year performed as **MLOps Senior Engineer** for a **Fintech** in **LATAM**.



Non-academic Experience



- **PyCon Colombia** and **Python Bogotá co-organizer**.
Collaborations in ScipyLATAM and Jupyter LATAM.
- +3 years performed as **Software Engineer** for several **tech companies** in Colombia. /
- 3 years performed as **Technical Leader** of **Machine Learning** and **Data Science** in an USA startup.
- +1 year performed as **MLOps Senior Engineer** for a **Fintech** in LATAM.



Non-academic Experience



- **PyCon Colombia** and **Python Bogotá co-organizer**.
Collaborations in ScipyLATAM and Jupyter LATAM.
- +3 years performed as **Software Engineer** for several **tech companies** in Colombia.
- 3 years performed as **Technical Leader** of **Machine Learning and Data Science** in an USA startup.
- +1 year performed as **MLOps Senior Engineer** for a **Fintech** in LATAM.



Non-academic Experience



- **PyCon Colombia** and **Python Bogotá co-organizer**.
Collaborations in ScipyLATAM and Jupyter LATAM.
- +3 years performed as **Software Engineer** for several **tech companies** in Colombia.
- 3 years performed as **Technical Leader** of **Machine Learning and Data Science** in an USA startup.
- +1 year performed as **MLOps Senior Engineer** for a **Fintech** in LATAM.



Outline

- 1 You don't know who I am
- 2 **Course Overview**
- 3 Syllabus
- 4 Grading & Rules
- 5 Bibliography



Overview

This course is designed to **introduce undergraduate students** to the **object-oriented design** as part of the foundation for becoming an experienced both software developer and software architect.

The course starts with a **brief introduction** to **object-oriented thinking**, **quality attributes**, and **software modeling**. Then, it transitions into **object-oriented analysis and design**, **modular design principles**. Finally, we will focus in **object-oriented programming**, **good practices**, and basic **UML Diagrams** and **Documentation**.

Classes will consist of **lectures**, **discussions**, and **practical examples**. Also, you must take some readings from **software engineering**. In addition, there will be a **semester-long project**, as well **one course final test**, **four workshops**, and **ten additional assignments**.



Overview

This course is designed to **introduce undergraduate students** to the **object-oriented design** as part of the foundation for becoming an experienced both *software developer* and *software architect*.

The course starts with a **brief introduction** to **object-oriented thinking**, **quality attributes**, and **software modeling**. Then, it transitions into **object-oriented analysis and design**, including **design principles**. Finally, we will focus in **object-oriented programming**, **good practices**, and basic **UML Diagrams** and **Documentation**.

Classes will consist of **lectures**, **discussions**, and **practical examples**. Also, you must take some readings from *software engineering*. In addition, there will be a **semester-long project**, as well **one course final test**, **four workshops**, and **ten additional assignments**.



Overview

This course is designed to **introduce undergraduate students** to the **object-oriented design** as part of the foundation for becoming an experienced both *software developer* and *software architect*.

The course starts with a **brief introduction** to **object-oriented thinking**, **quality attributes**, and **software modeling**. Then, it transitions into **object-oriented analysis and design**, including **design principles**. Finally, we will focus in **object-oriented programming**, **good practices**, and basic **UML Diagrams** and **Documentation**.

Classes will consist of **lectures**, **discussions**, and **practical examples**. Also, you must take some readings from *software engineering*. In addition, there will be a **semester-long project**, as well **one course final test**, **four workshops**, and **ten** additional **assignments**.

Backward + GUI



Goals

The **main goal** of this course is to **provide you** with **different concepts**, **tips**, **models** and **tools** for solving **software problems** using **object-oriented paradigm**.

At the end of this course you should be able to **create** a **full-software monolithich project solution** with a **good level of quality**. Also, you should be able to **design** robust **software systems** based on **object-oriented paradigm** in an **agnostic** way.



Goals

The **main goal** of this course is to **provide you** with **different concepts, tips, models and tools** for solving **software problems** using **object-oriented paradigm**.

At the end of this course you should be able to create a full-software monolithic project solution with a good level of quality. Also, you should be able to design robust software systems based on object-oriented paradigm in an agnostic way.



Pre-Requisites

This is a basic course, so you must have some knowledge in:

- **Programming** in Java, Python, or C++.

Also, it is desirable that you have some knowledge in:

- Git basic usage and its basic usage.
- Use of IDEs like VS Code, Eclipse, or PyCharm.

function
conditionals



Pre-Requisites

This is a basic course, so you must have some knowledge in:

- **Programming** in [Java](#), [Python](#), or [C++](#).

Also, it is desirable that you have some knowledge in:

- **Git** [basic usage](#), and **GitHub** [basic usage](#).
- Use of **IDEs** like [VS Code](#), [Eclipse](#), or [PyCharm](#).

account

copilot



Outline

- 1 You don't know who I am
- 2 Course Overview
- 3 Syllabus**
- 4 Grading & Rules
- 5 Bibliography



Syllabus I

Period	Topic	Time
Period I	Object-Oriented Analysis & Design	5 sessions
	Object-Oriented Modeling	6 sessions
	Workshop on Object-Oriented Design	1 session
	OOP — Inheritance and Polymorphism	10 sessions
	Workshop on OOP Implementation	1 session
	Course Project Catch-Up	1 session

Table: Schedule for Period I

Overleaf

Paper IEEE, LaTeX
Poster report



Syllabus II

Period	Topic	Time
Period II	Object-Oriented Principles	11 sessions
	Workshop on OOP Principles	1 session
	Layer Architectures	8 sessions
	Concurrency	2 sessions
	Workshop on Layer Architectures	1 session
	Final Test	1 session
Period III	Projects Dissertation	2 session

Table: Schedule for Period II & III



Outline

- 1 You don't know who I am
- 2 Course Overview
- 3 Syllabus
- 4 Grading & Rules**
- 5 Bibliography



Grades Percentages

Period	Item	Percentage
Period I	Assignments	5%
	Workshops	20%
	Project Catch-Up	10%
Period II	Assignments	5%
	Workshops	20%
	Test	10%
Period III	Paper + Poster	5%
	Project Implementation	15%
	Technical Report	10%

1%
 10%
 1%
 10%
 10%
 40%

Table: Software Modeling Grades Distribution



Don't hate the player, hate the game

- All assignments must be submitted **hand-written** on **time** and in **english**. Grammar and spelling will **not** be evaluated.
- Copying and pasting from internet is **forbidden**. Please, develop your own both ideas and solutions. **200-250**
- Class attendance is **not mandatory**. If you miss classes, you must study by yourself. **Name**
Task #
- No cell phones, no smartwatches, no whatsapp, no tinder, no smart-anything. **Course words**
Just you and your brain. Pay attention at clase.
- Communications with me must be done by email or by slack. I will not answer any question by WhatsApp.



Don't hate the player, hate the game

- All assignments must be submitted **hand-written** on **time** and in **english**. Grammar and spelling will **not** be evaluated.
- **Copying** and **pasting** from internet is **forbidden**. Please, **develop** your own both ideas and solutions.
- Class attendance is **not mandatory**. If you **miss** classes, you must *study by yourself*.
- No cell-phones, no smartwatches, no whatsapp, no tinder, no smart-anything. **Just you and your brain**. Pay attention at clase.
- Communications with **L**me must be done by **email** or by **slack**. I will **not** answer any question by *WhatsApp*.



Don't hate the player, hate the game

- All assignments must be submitted **hand-written** on **time** and in **english**. Grammar and spelling will **not** be evaluated.
- Copying and pasting from internet is **forbidden**. Please, **develop** your **own both ideas and solutions**.
- Class attendance is **not mandatory**. If you **miss** classes, you must **study by yourself**.
- No cell-phones, no smartwatches, no whatsapp, no tinder, no smart-anything. **Just you and your brain**. Pay attention at clase.
- Communications with me must be done by **email** or by **slack**. I will **not** answer any question by *WhatsApp*.



Don't hate the player, hate the game

- All assignments must be submitted **hand-written** on **time** and in **english**. Grammar and spelling will **not** be evaluated.
- Copying and pasting from internet is **forbidden**. Please, **develop** your own both ideas and solutions.
- Class attendance is **not mandatory**. If you **miss** classes, you must *study by yourself*.
- No cell-phones, no smartwatches, no whatsapp, no tinder, no smart-anything. **Just you and your brain**. Pay attention at clase.
- Communications with me must be done by email or by slack. I will not answer any question by WhatsApp.



Don't hate the player, hate the game

- All assignments must be submitted **hand-written** on **time** and in **english**. Grammar and spelling will **not** be evaluated.
- Copying and pasting from internet is **forbidden**. Please, **develop** your **own both ideas and solutions**.
- Class attendance is **not mandatory**. If you **miss** classes, you must *study by yourself*.
- No cell-phones, no smartwatches, no whatsapp, no tinder, no **smart-anything**. **Just you and your brain**. **Pay attention at clase**.
- Communications with me must be done by email or by slack. I will **not** answer any question by WhatsApp.

@chamel



Code of Conduct

- **Always** be **respectful** to your classmates and to me. You must be **kind** with everyone inside (*and outside*) the classroom.
- There is **no** a better programming language, tool, or technology. There are only **better** or **worse** solutions.
- You must be **honest** with your work. If you **don't know something**, just **ask** me. I will be **glad** to help you.
- You must be **responsible** with your work. If you don't submit **on time**, please **don't cry**.
- You must **not be annoying**, or affect the **classroom environment**. If you do, I will **ask you to leave** the classroom.



Code of Conduct

- Always be **respectful** to your **classmates** and to me. You must be **kind** with everyone inside (~~and outside~~) the classroom.
- There is **no** a better **programming language**, **tool**, or **technology**. There are only **better** or **worse** solutions.
- You must be **honest** with your work. If you **don't know something**, just **ask me**. I will be **glad** to help you.
- You must be **responsible** with your work. If you don't submit **on time**, please **don't cry**.
- You must **not be annoying**, or affect the **classroom environment**. If you do, I will **ask you to leave** the classroom.



Code of Conduct

- Always be **respectful** to your **classmates** and to me. You must be **kind** with everyone inside (*and outside*) the classroom.
- There is **no** a better **programming language**, **tool**, or **technology**. There are only **better** or **worse** solutions.
- You must be **honest** with your work. If you **don't know something**, just **ask** me. I will be **glad** to help you.
- You must be **responsible** with your work. If you don't submit **on time**, please **don't cry**.
- You must **not be annoying**, or affect the **classroom environment**. If you do, I will **ask you to leave** the classroom.



Code of Conduct

- Always be **respectful** to your **classmates** and to me. You must be **kind** with everyone inside (*and outside*) the classroom.
- There is **no** a better **programming language**, **tool**, or **technology**. There are only **better** or **worse** solutions.
- You must be **honest** with your work. If you **don't know something**, just **ask** me. I will be **glad** to help you.
- You must be **responsible** with your work. If you **don't submit on time**, please **don't cry**.
- You must **not be annoying**, or affect the **classroom environment**. If you do, I will **ask you to leave** the classroom.



Code of Conduct

- Always be **respectful** to your **classmates** and to me. You must be **kind** with everyone inside (*and outside*) the classroom.
- There is **no** a better **programming language**, **tool**, or **technology**. There are only **better** or **worse** solutions.
- You must be **honest** with your work. If you **don't know something**, just **ask** me. I will be **glad** to help you.
- You must be **responsible** with your work. If you don't submit **on time**, please **don't cry**.
- You must **not be annoying**, or affect the **classroom environment**. If you do, I will **ask you** to **leave** the classroom.



Outline

- 1 You don't know who I am
- 2 Course Overview
- 3 Syllabus
- 4 Grading & Rules
- 5 Bibliography**



Bibliography

Recommended bibliography:

- **Construcción de Software Orientado a Objetos**, by [Bertrand Meyer](#).
- **Thinking Java**, by [Bruce Eckel](#).
- **Java2: How To Program**, by [Deitel & Deitel](#).
- **Object-Oriented Analysis and Design**, by [Grady Booch](#).
- **Design Patterns: Elements of Reusable Object-Oriented Software**, by [Erich Gamma](#), [Richard Helm](#), [Ralph Johnson](#), & [John Vlissides](#).



Outline

- 1 You don't know who I am
- 2 Course Overview
- 3 Syllabus
- 4 Grading & Rules
- 5 Bibliography



Thanks!

Questions?



My Profile: www.linkedin.com/in/casierrav

