

ADVANCED PROGRAMMING I

Course Description

Author: Eng. Carlos Andrés Sierra, M.Sc.
`carlos.andres.sierra.v@gmail.com`

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

2024-I



UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS

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**, M.Sc. in Computer Engineering, *researcher* for 15 years.
- 7 years as **full-time associate 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**, M.Sc. in Computer Engineering, *researcher* for 15 years.
- 7 years as **full-time associate 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**, M.Sc. in Computer Engineering, *researcher* for 15 years.
- 7 years as **full-time associate 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**, M.Sc. in Computer Engineering, *researcher* for 15 years.
- 7 years as **full-time associate 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 as **software engineer** for several companies in Colombia.
- 3 years as **Technical Leader** of Machine Learning and Data Science in a USA startup.
- 1 year as **MLOps Engineer** for a Fintech in LATAM.



Non-academic Experience



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



Non-academic Experience



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



Non-academic Experience



- PyCon Colombia and Python Bogotá **co-organizer**. Collaborations in ScipyLATAM and Jupyter LATAM.
- 3 years as **software engineer** for several companies in Colombia.
- 3 years as **Technical Leader** of Machine Learning and Data Science in a USA startup.
- 1 year as **MLOps 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 some advanced topics of **object-oriented modeling** and *good practices* of code implementation. This is **not** a course fully focus on **software architecture**, but it is part of main concepts of software achitecture.

Classes will consist of lectures, **discussions**, practical examples, and workshops. Also, you must take some readings from *software architecture*. In addition, there will be a **semester-long project**, as well **two** exams, **one** paper, **eight** workshops, and **thirty** additional assignmens.



Overview

This course is designed to introduce undergraduate students to some advanced topics of **object-oriented modeling** and *good practices* of code implementation. This is **not** a course fully focus on **software architecture**, but it is part of main concepts of software achitecture.

Classes will consist of lectures, **discussions**, practical examples, and workshops. Also, you must take some readings from *software architecture*. In addition, there will be a **semester-long project**, as well **two** exams, **one** paper, **eight** workshops, and **thirty** additional assignmens.



Goals

The main goal of this course is to provide undergraduate students with different **models** and **tools** for solving software problems using **object-oriented design**.

At the end of this course you should be able to **create** a software **backend solution** with a good level of **quality** metrics. Also, you should be able to **design** robust software systems in an **agnostic** way.



Goals

The main goal of this course is to provide undergraduate students with different **models** and **tools** for solving software problems using **object-oriented design**.

At the end of this course you should be able to **create** a software **backend solution** with a good level of **quality** metrics. Also, you should be able to **design** robust software systems in an **agnostic** way.



Prerequisites

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

- **Programming** in Java, Python, or C++.
- Object-Oriented Programming foundations.
- UML and Class Diagrams basic concepts.
- Git basic usage, and GitHub basic usage.
- Data systems and relational model basic concepts.
- Use of IDEs like VS Code, Eclipse, or PyCharm.



Prerequisites

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

- **Programming** in Java, Python, or C++.
- **Object-Oriented Programming** foundations.
- UML and **Class Diagrams** basic concepts.
- **Git** basic usage, and **GitHub** basic usage.
- **Data systems** and relational model basic concepts.
- Use of **IDEs** like VS Code, Eclipse, or PyCharm.



Prerequisites

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

- **Programming** in Java, Python, or C++.
- **Object-Oriented Programming** foundations.
- UML and **Class Diagrams** basic concepts.
- **Git** basic usage, and **GitHub** basic usage.
- **Data systems** and relational model basic concepts.
- Use of **IDEs** like VS Code, Eclipse, or PyCharm.



Prerequisites

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

- **Programming** in Java, Python, or C++.
- **Object-Oriented Programming** foundations.
- UML and **Class Diagrams** basic concepts.
- **Git** basic usage, and **GitHub** basic usage.
- **Data systems** and relational model basic concepts.
- Use of **IDEs** like VS Code, Eclipse, or PyCharm.



Prerequisites

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

- **Programming** in Java, Python, or C++.
- **Object-Oriented Programming** foundations.
- UML and **Class Diagrams** basic concepts.
- **Git** basic usage, and **GitHub** basic usage.
- **Data systems** and relational model basic concepts.
- Use of **IDEs** like VS Code, Eclipse, or PyCharm.



Prerequisites

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

- **Programming** in Java, Python, or C++.
- **Object-Oriented Programming** foundations.
- UML and **Class Diagrams** basic concepts.
- **Git** basic usage, and **GitHub** basic usage.
- **Data systems** and relational model basic concepts.
- Use of **IDEs** like VS Code, Eclipse, or PyCharm.



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 Programming	3 classes
	Workshop: Classes in Python	1 session
	UML and Class Diagrams	2 classes
	Workshop on UML Diagrams	1 session
	Inheritance, Abstraction and Polymorphism	2 classes
	Classes, Packages, and Spaces	1 class
	Workshop on Object-Oriented Relations	1 session
Period II	Paper Revision	1 session
	Object-Oriented Design	3 classes
	Workshop on Object-Oriented Design	1 session
	Resources, Memory, Serialization	2 classes
	Workshop on Resources Management	1 session
	Test 1	1 session

Table: Schedule for Period I & II



Syllabus II

Period	Topic	Time
Period III	UI with Python TKinter	3 classes
	Workshop on Python UI	1 session
	DataBases, DAOs, DTOs	1 class
	Workshop on PostgreSQL and SQLAlchemy	1 session
	Architecture on Layers and Monoliths	2 classes
	Workshop on Monoliths	1 session
	Questions and Answers	2 classes
	Final Test	1 session
	Projects Presentation	1 session

Table: Schedule for Period 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	10%
	Workshops	15%
	Paper	10%
Period II	Assignments	10%
	Workshops	15%
	Test	10%
Period III	Final Paper	5%
	Final Test	10%
	Course Project	15%

Table: Software Modeling Grades Distribution



Don't hate the player, hate the game

- All assignments must be hand-written submitted on **time** and in **english**. Grammar and spelling will **not** be evaluated.
- Copying and pasting from internet is **forbidden**. Please, **develop** your own 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 smartanything. **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 hand-written submitted on **time** and in **english**. Grammar and spelling will **not** be evaluated.
- Copying and pasting from internet is **forbidden**. Please, **develop** your own 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 smartanything. **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 hand-written submitted on **time** and in **english**. Grammar and spelling will **not** be evaluated.
- Copying and pasting from internet is **forbidden**. Please, **develop** your own 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 smartanything. **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 hand-written submitted on **time** and in **english**. Grammar and spelling will **not** be evaluated.
- Copying and pasting from internet is **forbidden**. Please, **develop** your own 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 smartanything. **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 hand-written submitted on **time** and in **english**. Grammar and spelling will **not** be evaluated.
- Copying and pasting from internet is **forbidden**. Please, **develop** your own 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 smartanything. **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*.



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:

- **Clean Code: A Handbook of Agile Software Craftsmanship**, by Robert C. Martin.
- **Refactoring: Improving the Design of Existing Code**, by Martin Fowler.
- **Construcción de Software Orientado a Objetos**, by Bertrand Meyer.
- **Thinking Java**, by Bruce Eckel.
- **Java2 How To Program**, by Deitel & Deitel.



Bibliography

Recommended bibliography:

- **Python 3 Object-Oriented Programming**, by Dusty Phillips.
- **Fluent Python: Clear, Concise, and Effective Programming**, by Luciano Ramalho.
- **Effective Python: 90 Specific Ways to Write Better Python**, by Brett Slatkin.
- **Python Cookbook: Recipes for Mastering Python 3**, by David Beazley.



Outline

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



Thanks!

Questions?



www.linkedin.com/in/casierrav

