Computer Science I

Syllabus

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

Full-time Adjunct Professor
Computer Engineering Program
School of Engineering
Universidad Distrital Francisco José de Caldas

2026-I

Outline

Academic Experience

- **Computer Engineer**, M.Sc. in Computer Engineering, and *researcher* for 16 years.
- 8 years as **full-time associate professor** at colleges, in Computer Engineering programs.
- 3 years as **lecturer professor** for both colleges and government STEM programs.
- Speaker at IEEE events and colleges in Colombia, Brazil, and Bolivia.

UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS

Non-academic Experience

- PyCon Colombia and Python Bogotá **co-organizer**.
- 3 years as **software engineer** for several tech companies in Colombia.
- 3 years as **Technical Leader** of Machine Learning and Data Science at a USA startup.
- 1.5 years as **MLOps Engineer** for a Fintech company in LATAM.
- Currently, **Senior Engineering Manager** of Data Engineering and Machine Learning at `Blend` 360.

UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS

Outline

Overview

This course is designed to introduce undergraduate students to **algorithmic problem-solving** as part of the foundation for becoming an experienced *software engineer* capable of developing *efficient solutions*.

The course starts with a comprehensive analysis of **problem context** and **constraints identification**. Then, it transitions into **algorithmic design** and alternative solution approaches. Finally, we will focus on **complexity analysis**, **data structures**, and **optimization techniques** for both memory and time resources.

Classes will consist of lectures, **problem-solving sessions**, and practical implementations. Also, you must complete some readings from *algorithm analysis* and *data structures*. In addition, there will be a **semester-long project**, one **final course test**, and four **workshops**.

Overview

This course is designed to introduce undergraduate students to **algorithmic problem-solving** as part of the foundation for becoming an experienced *software engineer* capable of developing *efficient solutions*.

The course starts with a comprehensive analysis of **problem context** and **constraints identification**. Then, it transitions into **algorithmic design** and alternative solution approaches. Finally, we will focus on **complexity analysis**, **data structures**, and **optimization techniques** for both memory and time resources.

Classes will consist of lectures, **problem-solving sessions**, and practical implementations. Also, you must complete some readings from *algorithm analysis* and *data structures*. In addition, there will be a **semester-long project**, one **final course test**, and four **workshops**.

Overview

This course is designed to introduce undergraduate students to **algorithmic problem-solving** as part of the foundation for becoming an experienced *software engineer* capable of developing *efficient solutions*.

The course starts with a comprehensive analysis of **problem context** and **constraints identification**. Then, it transitions into **algorithmic design** and alternative solution approaches. Finally, we will focus on **complexity analysis**, **data structures**, and **optimization techniques** for both memory and time resources.

Classes will consist of lectures, **problem-solving sessions**, and practical implementations. Also, you must complete some readings from *algorithm analysis* and *data structures*. In addition, there will be a **semester-long project**, one **final course test**, and four **workshops**.

UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS

Goals

The main goal of this course is to present students with theoretical concepts and **practical applications** for **algorithm analysis** and computational problem-solving.

At the end of this course you should be able to **perform** computational complexity analysis of algorithms, expressing the **resource usage** in terms of mathematical functions. Also, you should be able to **determine** the optimal data structure that minimizes algorithmic complexity for **specific problems**, optimizing both algorithms and **information management** in software solutions.

Goals

The main goal of this course is to present students with theoretical concepts and **practical applications** for **algorithm analysis** and computational problem-solving.

At the end of this course you should be able to **perform** computational complexity analysis of algorithms, expressing the **resource usage** in terms of mathematical functions. Also, you should be able to **determine** the optimal data structure that minimizes algorithmic complexity for **specific problems**, optimizing both algorithms and **information management** in software solutions.

Pre-Requisites

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

- **Programming** in Java, Python, or C++.
- Basic **object-oriented programming** concepts.

Additionally, it is desirable that you have some knowledge of:

- Basic usage of **Git** and **GitHub**.
- Use of **IDEs** such as VS Code, Eclipse, or PyCharm.

Pre-Requisites

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

- **Programming** in Java, Python, or C++.
- Basic **object-oriented programming** concepts.

Additionally, it is desirable that you have some knowledge of:

- Basic usage of **Git** and **GitHub**.
- Use of **IDEs** such as VS Code, Eclipse, or PyCharm.

Outline

Syllabus I

Period	Topic	Time
Period I	Introduction to Algorithms	3 sessions
	Algorithms Design	3 sessions
	Algorithms Types and Paradigms	6 sessions
	Workshop: Ad-Hoc Problem Solving Contest	1 session
	Search Algorithms	3 sessions
	Sorting Algorithms	6 sessions
	Complexity Analysis I	4 sessions
	Workshop on Sorting Algorithms	1 session
	Course Project Catch-Up	2 sessions

Table: Schedule for Period I

Syllabus II

Period	Topic	Time
Period II	Complexity Analysis II	5 sessions
	Linear Data Structures	9 sessions
	Workshop on Linear Data Structures	1 session
	Tree Data Structures	9 sessions
	Workshop on Tree Data Structures	1 session
	Final Test	1 session
Period III	Project Dissertation	2 session

Table: Schedule for Period II & III

Outline

UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS

Grades Percentages

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

Don't hate the player, hate the game

- All assignments must be submitted handwritten, **on time**, and in **English**. Grammar and spelling will **not** be evaluated.
- Copying and pasting from the internet are **forbidden**. Please **develop** your own ideas and solutions.
- Class attendance is **not mandatory**. If you **miss** classes, you must study *independently*.
- No cell phones, smartwatches, WhatsApp, Tinder, or smart devices. **Just you and your brain**. Pay attention in class.
- Communication with me must be via **email** or **Slack**. I will **not** answer any questions via *WhatsApp*.

UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS

Don't hate the player, hate the game

- All assignments must be submitted handwritten, **on time**, and in **English**. Grammar and spelling will **not** be evaluated.
- Copying and pasting from the internet are **forbidden**. Please **develop** your own ideas and solutions.
- Class attendance is **not mandatory**. If you **miss** classes, you must study *independently*.
- No cell phones, smartwatches, WhatsApp, Tinder, or smart devices. **Just you and your brain**. Pay attention in class.
- Communication with me must be via **email** or **Slack**. I will **not** answer any questions via *WhatsApp*.

Don't hate the player, hate the game

- All assignments must be submitted handwritten, **on time**, and in **English**. Grammar and spelling will **not** be evaluated.
- Copying and pasting from the internet are **forbidden**. Please **develop** your own ideas and solutions.
- Class attendance is **not mandatory**. If you **miss** classes, you must study *independently*.
- No cell phones, smartwatches, WhatsApp, Tinder, or smart devices. **Just you and your brain**. Pay attention in class.
- Communication with me must be via **email** or **Slack**. I will **not** answer any questions via *WhatsApp*.

Don't hate the player, hate the game

- All assignments must be submitted handwritten, **on time**, and in **English**. Grammar and spelling will **not** be evaluated.
- Copying and pasting from the internet are **forbidden**. Please **develop** your own ideas and solutions.
- Class attendance is **not mandatory**. If you **miss** classes, you must study *independently*.
- No cell phones, smartwatches, WhatsApp, Tinder, or smart devices. **Just you and your brain**. Pay attention in class.
- Communication with me must be via **email** or **Slack**. I will **not** answer any questions via *WhatsApp*.

Don't hate the player, hate the game

- All assignments must be submitted handwritten, **on time**, and in **English**. Grammar and spelling will **not** be evaluated.
- Copying and pasting from the internet are **forbidden**. Please **develop** your own ideas and solutions.
- Class attendance is **not mandatory**. If you **miss** classes, you must study *independently*.
- No cell phones, smartwatches, WhatsApp, Tinder, or smart devices. **Just you and your brain**. Pay attention in class.
- Communication with me must be via **email** or **Slack**. I will **not** answer any questions via *WhatsApp*.

Code of Conduct

- Always be **respectful** to your classmates and to me. You must be **kind** to everyone inside (*and outside*) the classroom.
- There is no best 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 complain.
- You must not be **disruptive** or **negatively 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** to everyone inside (*and outside*) the classroom.
- There is no best 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 complain.
- You must not be **disruptive** or **negatively 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** to everyone inside (*and outside*) the classroom.
- There is no best 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 complain.
- You must not be **disruptive** or **negatively 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** to everyone inside (*and outside*) the classroom.
- There is no best 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 complain.
- You must not be **disruptive** or **negatively 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** to everyone inside (*and outside*) the classroom.
- There is no best 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 complain.
- You must not be **disruptive** or **negatively affect** the **classroom environment**. If you do, I will ask you to **leave** the classroom.

Outline

UNIVERSIDAD DISTRITAL FRANCISCO JOSÉ DE CALDAS

Bibliography

Recommended bibliography:

- **Introduction to Algorithms**, by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, & Clifford Stein.
- **The Algorithm Design Manual**, by Steven S. Skiena.
- **Data Structures and Algorithms in Java**, by Michael T. Goodrich, Roberto Tamassia, & Michael H. Goldwasser.
- **Algorithms**, by Robert Sedgewick & Kevin Wayne.
- **Data Structures and Algorithm Analysis in C++**, by Mark Allen Weiss.
- **Grokking Algorithms: An Illustrated Guide for Programmers and Other Curious People**, by Aditya Bhargava.

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

Thanks!

Questions?

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