Paris School of AI — PSAI

Bachelor Programme — IBSAI Year 1 (B1)

Computer Programming 1

Course Coordinator: Last Name / First Name: SENELLART Pierre
Course Coordinator Affiliation: Professor of Computer Science, ENS-PSL
Course Coordinator webpage: https://pierre.senellart.com/

Other Instructor(s) NOM / Prénom: WANG Changhong, SAADEH Angelo (tutorials, group 1); PATSILINAKOS Panagiotis (tutorials, group 2)
Other Instructor(s) Affiliation: PSL AI Fellows (Changhong Wang, Panagiotis Patsilinakos); ENS-PSL Research Fellow (Angelo Saadeh)
Other Instructor(s) webpage: https://changhongw.github.io/
https://www.lamsade.dauphine.fr/~ppatsilinakos/ https://www.angelosaadeh.fr/

Learning Objectives : Acquire the habit of computational thinking: decomposing a problem into steps, automating it, and making the solution faster. Become familiar with writing, reading, and debugging programs in Python and C, along with a fragment of C++, with a focus on principles and good practices that transfer across languages. Understand how data is represented and stored in memory, and how memory is managed. Choose appropriate data structures and evaluate the quality of an algorithm through its time and space complexity. Apply the common algorithmic techniques (recursion, divide and conquer, dynamic programming, greedy algorithms) and the classical sorting algorithms. Structure programs with functions, modules, and objects; handle exceptional situations and input/output robustly; specify, test, and debug programs; and get acquainted with the basic tools of software engineering (version control, functional and generic programming, regular expressions).

Course Description : An introduction to computer programming, algorithms, and data structures, instantiated in Python and C (with a subset of C++). The course starts from what a computer is and how it represents data, covers structured programming, collections, algorithmic complexity, memory management, object-oriented programming, and the common data structures and algorithmic techniques, and ends with the practical skills of the working programmer: exceptions, sorting, input/output, correctness and testing, functional and generic programming, version control with git, and regular expressions. Each lecture is put into practice in a hands-on tutorial on the students' own laptops.

Week #	Content
1	Introduction to computing: computers and programming languages (Python, C, C++), data representation, variables, expressions, basic input/output.
2	Structured programming: control flow statements, functions.
3	Quality of an algorithm: algorithmic complexity, O/Ω/Θ notation, time and space complexity, example complexity calculations.

Week #	Content
	Python collections.
4	Memory management, collections in C and C++, dynamic arrays, amortized complexity.
5	Object-oriented programming: concepts, implementation in Python and C++, structured data in C.
6	Common data structures for collections: ordered sequences, value-based collections, associative arrays. Multi-file programs: headers and implementation files in C/C++, modules and packages in Python.
7	Common algorithmic techniques: recursion, memoization, dynamic programming, greedy algorithms, divide and conquer.
8	**Midterm**. Dealing with exceptional situations: exceptions in C, C++, and Python.
9	Sorting algorithms: insertion sort, merge sort, quicksort, lower bound for comparison sorts, sorting without comparisons, heapsort.
10	Input and output: basic concepts, the operating system view, lines, tokens, and types, safety and robustness.
11	Building correct programs: specifications, reasoning about programs, testing, debugging.
12	Basics of functional and generic programming, in Python, C, C++.
13	Basics of version control with git. Regular expressions in practice: grep, Python, C++.
14	Practice exam and revision.

Language of Instruction: English.

Course Format: One class per week, consisting of a 3-hour lecture on Monday morning, starting with a 10-minute flash quiz, and a 3-hour tutorial (hands-on exercise session on the students' own laptops, in two groups) on Wednesday afternoon.

Total des heures pour le semestre en cours: 84h / semester (42h of lectures, 42h of tutorials).

Assessment Methods: Quizzes 20% (short quiz at the beginning of each lecture), mid-semester exam 35% (written, 1.5 hours, 6 November 2026), final exam 45% (written, 2 hours, 25 January 2027). Tutorials are not graded.

Level: Bachelor IBSAI, Year 1

Semester: Semester 1

Required Readings: None

Recommended Readings:

- Wassberg, *Computer Programming for Absolute Beginners*.
- Cormen, Leiserson, Rivest & Stein, *Introduction to Algorithms*.
- Lutz, *Learning Python: Powerful Object-Oriented Programming*.

- King, *C Programming: A Modern Approach*.
- Python documentation. https://docs.python.org/3/
- C and C++ reference. https://en.cppreference.com/
- Learn C++. https://www.learncpp.com/