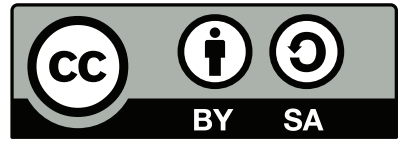
Scientific Programming

Course introduction

Andrea Passerini

Università degli Studi di Trento

2020/09/21



Organization

145540 Scientific Programming (12 ECTS, LM QCB)
145685 Scientific Programming (12 ECTS, LM Data Science)

Part A - Programming (22/9-29/10)

Introduction to the Python language and to a collection of programming libraries for data analysis.

- Mutuated as 145912 Scientific Programming (LM Math, 6 credits)

Part B - Algorithms (3/11-14/12)

Design and analysis of algorithmic solutions. Presentation of the most important classes of algorithms and evaluation of their performance.

Course syllabus - Part A

- Introduction to Python
- Data types
  - Numbers
  - Strings
  - Lists, tuples, dictionaries
- Input-Output
  - Raw input
  - File system
- Complex statements
  - If
  - For, while
  - Nested statements
- Functions
  - Definition
  - Calls
  - Return values
- Programs
  - Structuring a program
  - Importing external modules
- Libraries
  - Pandas
  - Numpy
  - MatPlotLib

Course syllabus - Part B

- Introduction
  - Recursion
  - Algorithm analysis
  - Asymptotic notations
- Data structures
  - High level overview
  - Sequences, maps (ordered/unordered), sets
  - Data structure implementations in Python
- Trees
  - Data structure definition
  - Visits
- Graphs
  - Data structure definition
  - Visits
  - Algorithms on graphs
- Algorithmic techniques
  - Divide-et-impera
  - Dynamic programming
  - Greedy
  - Backtrack
  - NP class: brief overview

Objectives of the course – Part A

At the end of the module, students are expected to:

- Remember the syntax and semantics of the Python language;
- Understand programs written by others individuals;
- Analyze a simple data analysis task and reformulate it as a programming problem;
- Evaluate which features of the language (and related scientific libraries) can be used to solve the task;
- Construct a Python program that appropriately solves the task;
- Evaluate the results of the program.

Objectives of the course – Part B

At the end of the module, students are expected to:

- evaluate algorithmic choices and select the ones that best suit their problems;
- analyze the complexity of existing algorithms and algorithms created on their own;
- design simple algorithmic solutions to solve basic problems.

What you will learn

Programming expertise

- Content: a brief overview of the main problems in algorithmics and their solution
- Approach: the principles and the techniques that can be used to solve such problems

Content: list of algorithms

- Read their code
- Understand why they work
- Try to implement them

Approach: abstract thinking

- Develop new solutions for unusual problems

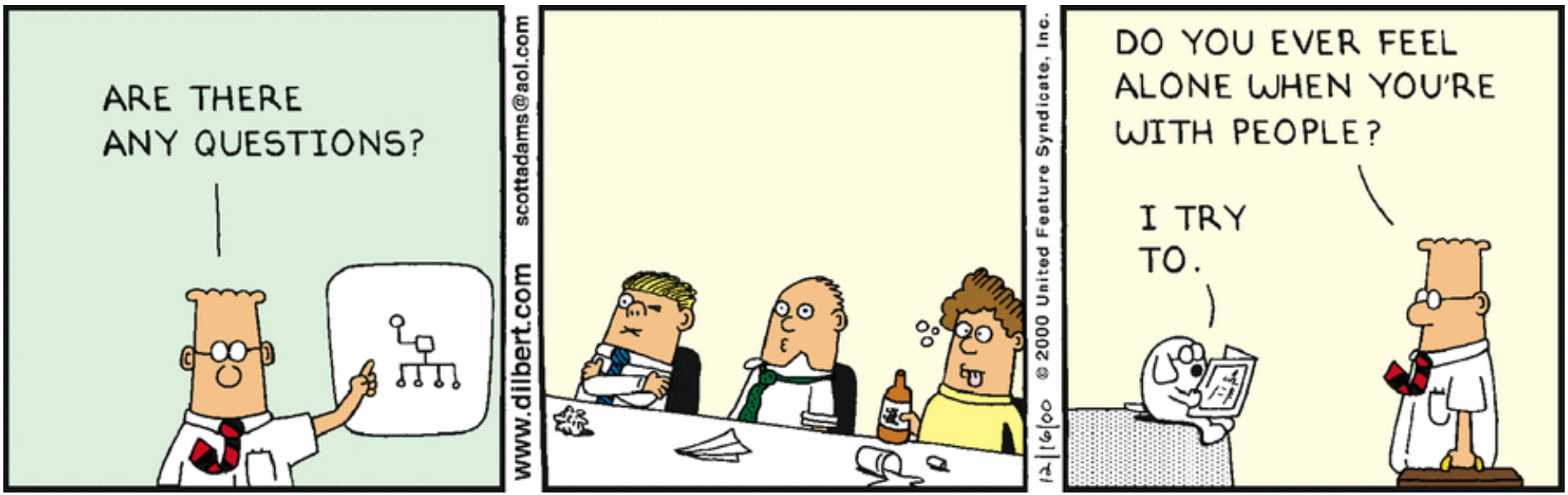
About interaction during the lecture

Ask questions!!

- If I'm not clear enough, stop me!
- If you want additional information, ask!

He who asks a question is a fool for a minute; he who does not ask a question remains a fool forever

Chinese proverb



Course material

`http://disi.unitn.it/~passerini/teaching/2020-2021/sci-pro/`

- Slides and notes (in development)
- Links to additional material

Moodle course page

- Communications
- Lecture recordings

Instructors: Part A - Programming

- Instructor: Prof. Andrea Passerini
  - Theory lectures, programming exercises
  - andrea.passerini [AT] unitn.it
- Teaching assistant: Dr. Luca Bianco
  - Python lab sessions (QCB)
  - luca.bianco [AT] fmach.it
- Teaching assistant: Dr. David Leoni
  - Python lab sessions (data science)
  - david.leoni [AT] unitn.it

Instructors: Part B - Algorithms

- Instructor: Dr. Luca Bianco
  - Theory lectures, algorithmic exercises
  - luca.bianco [AT] fmach.it
- Teaching assistant: Dr. Erik Dassi
  - Lab sessions on algorithms (QCB)
  - erik.dassi [AT] unitn.it
- Teaching assistant: Dr. David Leoni
  - Lab sessions on algorithms (data science)
  - david.leoni [AT] unitn.it

Schedule

Week day	**Time**	**Room**	**Description**
Monday	14.30-16.30	online	Lab
Tuesday	15.30-17.30	online	Lecture
Wednesday	11.30-13.30	online	Lab
Thursday	15.30-17.30	online	Lecture

Exam

145540,145685 Scientific Programming (12 credits)

- Lab exam
  - Python programming
  - Simple algorithmic problems
  - Questions about computational complexity

145912 Scientific Programming (6 credits, Math)

- Lab exam
  - Python programming

Dates

Midterms

Midterm 1 (1.5h)	4/11
Midterm 2 (1.5h)	16/12

Full exams

January (3h)	TBD
February (3h)	TBD
June (3h)	TBD
July (3h)	TBD
September (3h)	TBD

Mark Registration

145540,145685 Scientific Programming (12 credits)

- If you pass **both** midterm exams, you can register the mark
- The mark is computed as the average of the marks of the midterm exams, rounded up (e.g. $(25+26)/2 = 26$)
- To register your mark you need to enroll to one of the regular sessions (not the midterm ones).
- If you passed both midterm exams, enroll to a session and do not show up, we assume you want to register your mark

Mark Registration

continued

- If you passed both midterm exams, enroll to a session and do show up, this means that you are not happy with the mark and want to take the full exam. The result of the full exam will be your new mark, you cannot backtrack to the midterm mark.
- If you did not pass both midterm exams, you need to take the full exam at a regular session.
- After the mark of a regular session have been published, you have a week to refuse it, after which it will be registered (silent assent registration).

Mark Registration

145912 Scientific Programming (6 credits, Math)

- If you pass the midterm exam, you can register the mark by enrolling to a regular session.
- If you passed the midterm exam, enroll to a session and do not show up, we assume you want to register your mark
- If you passed the midterm exam, enroll to a session and do show up, this means that you are not happy with the mark and want to take the exam again. The result of the regular session exam will be your new mark, you cannot backtrack to the midterm mark.
- After the mark of a regular session have been published, you have a week to refuse it, after which it will be registered (silent assent registration).