SUPSI

Introduction to the course

Massimo Coluzzi

Parallel and Concurrent Programming
Bachelor in Data Science

Parallel and Concurrent Programming (PaCP)

- This module will cover the spring semester.

- It will last 15 weeks:
    - o Every week will cover
        - 2 hours of theoretical lectures
        - 2 hours of practical exercises
    - o 4 hours of individual study beyond classroom hours

- lecturer:
  Massimo Coluzzi: <u>`massimo.coluzzi@supsi.ch`</u>

Lectures timesheet

D2A D2A-DATA-SCIENCE

	lunedì	martedì	mercoledì	giovedì	venerdì
8:30 9:15		C-D4201E D1.09 WAN Unsupervised learning	C-I4202E D1.09 GUI Software		L-I4202E D1.09 GUI Lab. Software Engineering
9:15 10:00					L-I4202E D1.09 OLD Lab. Software Engineering
10:15 11:00	A ? Asincrone	E-D4201E D1.09 WAN Ex. Unsupervised Learning		C-D4202E D1.09 PIG Industrial and Societal Applic	
11:00 11:45		C-D4204E D1.09 AYO Trustworthy Autonomous Systems	C-I4204E D1.09 CLZ Parallel Programming	E-D4202E D1.09 PIG Ex. Industrial and Societal Ap	E-B4205E D1.09 MRZ Ex. Elements of Signal Theory
11:45 12:30				E-B4202E D1.09 SAL Ex. Applied Operations Researc	
12:45 13:30		E-D4204E D1.09 AYO Ex. Trustworthy Autonomous Sys			E-P4203E D1.09 AZM Ex. Data Projects & Hackaton 2
13:30 14:15			C-B4205E D1.09 DNN Elements of Signal Theory and	C-B4302Z D1.09 RIZ Sistemi dinamici discreti	
14:30 15:15		E-I4204E D1.09 CLZ Ex. Parallel and Concurrent Pr		C-G4306Z B1.08 LND Industry 4.0	
15:15 16:00			C-B4202E D1.09 SAL Applied Operations Research		
16:00 16:45					
16:45 17:30					

Doubts or questions?

In case of doubts or questions you can leverage the following options:

- Write to the course instructor by email. To ask questions please do not use Teams' chat (outside class hours).

- Tutoring by the students, by appointment:
  http://form-dti.app.supsi.ch/form/view.php?id=215544

Course Objectives

The main objectives of this course are:

- Understand the principles of parallel and concurrent programming in a language independent way.

- Know how to apply such techniques in an operating environment and programming language.

- Know how to develop concurrent applications based on shared memory, message passing, synchronous and asynchronous events.

- Understand the basics of modern parallelization infrastructures. Study classical parallelization problems.

Course content

The main contents of this course are:

- Theoretical concepts of concurrency and synchronization: atomicity, visibility, thread-safety, liveness, load balancing and scalability.

- Elements of concurrent programming: thread, lock.

- Scheduling strategies with threads.

- Principles of parallel programming: models and techniques for parallel processing. Solutions for parallel programming in Python.

- Classical synchronization problems.

What this course is NOT about

This course in NOT about:

- Low level pthreads (POSIX threads) programming.

- Reactive programming.

- Python data science libraries (NumPy, Pandas, TensorFlow, PyTorch, ...).

- Distributed architectures (job queues SQS, RabbitMQ, Celery,...).

- Pipelining, clustering and distributed computing.

Teaching materials

- Lectures slides
- Examples and exercises

- Main reference books:
  - ➢ Packt: Learning Concurrency in Python (Elliot Forbes)
  - ➢ Packt: Mastering Concurrency in Python (Quan Nguyen)

- Other reference books:
  - ➢ Manning: Python Concurrency with asyncio (Matthew Fowler)
  - ➢ Packt: Modern Python Standard Library Cookbook (Alessandro Molina)
  - ➢ The Python 3 Standard Library by Example, Doug Hellmann, Addison-Wesley Professional

- The Python Standard Library
- Further references on iCorsi and on the slides.

Teaching materials

!!!
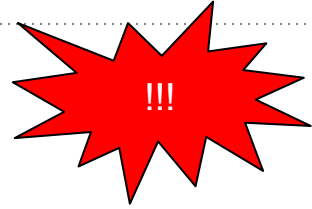
You will find lots of material on the internet and on books about the course topics. Be aware of small changes in the syntax performed in the different Python versions.

Tools and applications

- Personal laptops to have during all exercises.
- Lab computers used during written tests.

- ~~Microsoft Teams (for online lectures)~~
- Python
- Visual Studio Code
- Command line terminal

Python version

For this course we will use Python version 3.13.
We will use a containerized environment through devcontainer.

Notes about the slides

I am used to marking some slides with stamps (like slide number 9 in this presentation). The stamps aim to make you aware that such slides have a special meaning or content. The most frequently used stamps are the following:
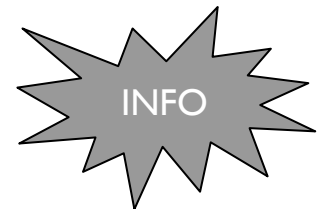
RECAP	AT HOME	!!!	INFO
The slide contains content that you should already know.	The slide contains examples or exercises that you should try at home.	The slide contains some important notes or content you should be aware of.	The slide contains additional content not part of the program and not evaluated.

Module grading

Grading method:

- Grades from 1 to 6 with decimal point.
- To pass this course you need to get at least a 4.

- There will be two evaluation sessions:
  - A mid-term test
  - The final exam

- The final grade will be calculated as follows:

```
module grade = mid-term test * 0.4 + final exam * 0.6
```

Note: Active participation in class can influence the final grade.

Absence to mid-term test

Absence to mid-term test:

- Non-justified absences to the test: the test grade is 1! This will be averaged with the other coursework elements in the composition of the module grade. However, if there is any doubt about the overall assessment of the module, the lecturer reserves the right to make the student take an additional certification test.

- The written justification for the absences must be hand to the secretariat, with copy to the lecturer.

- Justified absences to the test: the test must be recovered with an oral test in the following days. The exact time slot must be arranged with the lecturer.

Mid-term test

The expected date for the mid-term test is the following:

Wednesday 6th May 2025, during the exercise lecture.

Date could still be subject to change based on interactions with other courses.

- The mid-term test will be held in presence.
- The only material allowed during the written test is an A4 sheet (front and back) with a handwritten summary.

More detailed information about the mid-term test will follow during the course.

Lab - exercises

Exercise lecture:

- Labs will be held in presence.
- The attendance to the lab (exercises) lectures is mandatory.
- A set of exercises will be proposed each week. For each problem set you will have one week to develop and submit your solution.

Assignments

- The assignments will not be graded and corrected by the teacher but they will be taken into account for the final grade of the course.

- It is possible to ask for feedbacks about the assignments during the lab hours.

- It is possible to submit partial assignments.

- Group assignments must be reported in the header of your solution.

Assignments

- The assignment must be submitted within the established deadline. Delayed assignments submission will not be taken into consideration.

- A proposed solution will be loaded on the iCorsi platform in the days following the submission deadline.

- It is student's responsibility to view the proposed solutions for all exercises.

Pandemonium

- Furthermore, throughout the course, there will be an opportunity to create a small game named Pandemonium.

- The game's development is divided into 4 phases, from a sequential approach to a fully parallel approach.

- The game-related assignments are optional. But whoever delivers all 4 phases of the game will get an additional 5% on the final note.
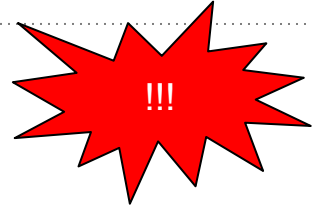
Importance of practicing

!!!

- Practice hours are mandatory.

- Practical questions or exercises may appear during the final test.

As Confucio used to say:

"If I listen I forget,
 if I read I remember,
 if I practice I understand."

General rules for remote lectures

- For the remote lectures we will use Microsoft Teams.

- The Teams channel of the course is the following:

    <u>DTI 2023 - M-I4070E.1 - Parallel and Concurrent Programming</u>

    Please join the team ASAP!

General rules for remote lectures

- When joining a remote lecture, activate the webcam in order to maintain visual contact. Webcam off is allowed only in case of technical problems.

- Keep your microphone off to avoid background noise. You will turn it on just when you need to speak.

- The behaviour during remote lectures must remain the same as that held in class: appropriate clothing, no smoking, no eating, no disappearing in the middle of the lesson, no playing with devices.

General rules for remote lectures

In case of questions during the lecture, use the "Raise hand" feature present in Teams and wait for the lecturer to give you word.

When participating to a meeting, a control bar with the main operations is available on your window. There you can find the "Raise hand" button.
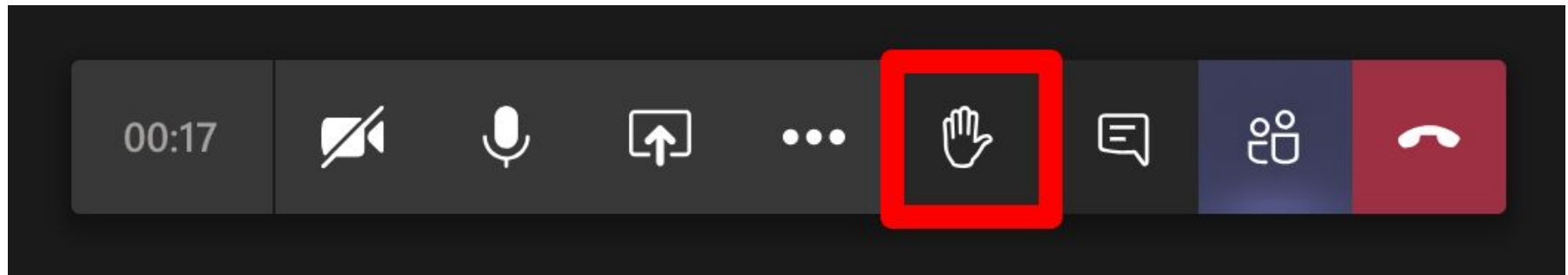
When finished with your question, please turn your microphone off and use the same button to lower your hand.

General rules for remote lectures

Any video or audio recording of a lecture needs the allowance of all participants (lecturer and students), both online and in presence.

This means that it is not allowed to record the lecturer without his knowledge (through the tools available in Teams or Zoom, or even by screen capture). Also the lecturer must ask to all students in case he wants to record a session.

The remote lectures are held in a delimited educational environment (accessible only with personal accounts), only for didactic purposes. Any recording must remain in this environment and it is thus illegal to share recordings on other channels and/or for other purposes, in particular when data is transmitted abroad (for example, through the services of social networks).

iCorsi

All the resources are available on iCorsi:

```
Corsi bachelor DATA SCIENCE AND ARTIFICIAL INTELLIGENCE
M-I4070E.1 - Parallel and Concurrent Programming (2024)

               Direct Link:
    https://www.icorsi.ch/course/view.php?id=20156
```

Official communications

All the communications related to the course will be done using the forum on the platform. It is student's responsibility to check regularly for possible announcements.