

## Lecture 00

Lect. PhD.  
Arthur Molnar

Introduction  
to course  
Schedule  
Objectives  
Course content  
Bibliography  
Activity and  
grading

# Introduction to Course

Lect. PhD. Arthur Molnar

Babes-Bolyai University

# Overview

## Lecture 00

Lect. PhD.  
Arthur Molnar

Introduction  
to course

Schedule

Objectives

Course content

Bibliography

Activity and  
grading

## 1 Introduction to course

- Schedule
- Objectives
- Course content
- Bibliography
- Activity and grading

# Guiding professors

## Lecture 00

Lect. PhD.  
Arthur Molnar

### Introduction to course

Schedule  
Objectives  
Course content  
Bibliography  
Activity and  
grading

- Lect. PhD. Arthur Molnar
- Lect. PhD. Radu Gaceanu
- Lect. PhD. Mircea Ioan-Gabriel
- Lect. PhD. Andrei Mihai
- Assist. Briciu Anamaria
- Assist. Imre Zsigmond
- Assist. Sergiu Nistor

# Schedule

## Lecture 00

Lect. PhD.  
Arthur Molnar

Introduction  
to course  
**Schedule**  
Objectives  
Course content  
Bibliography  
Activity and  
grading

- **Lecture:** 2 hours/week (online)
- **Seminar:** 2 hours/week (physical presence/1 group online)
- **Laboratory:** 2 hours/week (physical presence/1 subgroup online)
- **Consultation:** optional, each teacher has a weekly time slot (will be announced on Teams)

## Course materials

- **Teams, General** channel, **Files** section
- **FP** repository on GitHub Classroom

## Contact us

Best way is using **Teams** chat

# Objectives

## Lecture 00

Lect. PhD.  
Arthur Molnar

Introduction  
to course  
Schedule  
**Objectives**  
Course content  
Bibliography  
Activity and  
grading

## What should you gain from this course?

- Learn key programming concepts
- Learn the basic concepts of software engineering (design, implementation and maintenance of software systems)
- Learn to use basic software tools such as IDE's, documentation generators, testing tools
- Acquire and improve your programming style.
- Learn the basics of programming using the Python language

# Course content

## Lecture 00

Lect. PhD.  
Arthur Molnar

Introduction  
to course

Schedule

Objectives

**Course content**

Bibliography

Activity and  
grading

## How is this course organized?

- Programming in the large
- Programming in the small

# Programming in the large

## Lecture 00

Lect. PhD.  
Arthur Molnar

Introduction  
to course

Schedule  
Objectives

**Course content**

Bibliography  
Activity and  
grading

- 1 Procedural programming
- 2 Modular Programming
- 3 Test Driven Development
- 4 Design Principles for Modular Programs
- 5 User Defined Types and Exceptions
- 6 Introduction to UML
- 7 Design Principles for Object Oriented Programs
- 8 Program Testing. Refactoring.
- 9 Layered architecture. Inheritance.
- 10 Intro to building GUIs

# Programming in the small

## Lecture 00

Lect. PhD.  
Arthur Molnar

Introduction  
to course

Schedule

Objectives

**Course content**

Bibliography

Activity and  
grading

- 11** Recursion
- 12** Computational complexity
- 13** Searching. Sorting
- 14** Problem solving methods



# Bibliography

## Lecture 00

Lect. PhD.  
Arthur Molnar

Introduction  
to course  
Schedule  
Objectives  
Course content  
Bibliography  
Activity and  
grading

- 1 Kent Beck - **Test Driven Development: By Example**; Addison-Wesley Longman, 2002.
- 2 Kleinberg and Tardos **Algorithm Design**; Pearson Educational; 2014  
(<http://www.cs.princeton.edu/wayne/kleinberg-tardos/>)
- 3 Martin Fowler - **Refactoring. Improving the Design of Existing Code**; Addison-Wesley, 1999.  
(<http://refactoring.com/catalog/index.html>)
- 4 Frentiu, M., H.F. Pop, Serban G. - **Programming Fundamentals**; Cluj University Press, 2006
- 5 Online Python resources -  
<https://docs.python.org/3/reference/index.html>,  
<https://docs.python.org/3/library/index.html>,  
<https://docs.python.org/3/tutorial/index.html>

# Activity and grading

## Lecture 00

Lect. PhD.  
Arthur Molnar

Introduction  
to course  
Schedule  
Objectives  
Course content  
Bibliography  
Activity and  
grading

- **40%** - Laboratory work (assignments and tests) (**L**)
- **20%** - Written exam (during exam session) (**W**)
- **40%** - Practical test (during exam session) (**T**)
- **0 - 0.5p** Seminar activity (bonus to laboratory grade)
- **0 - 1p** Additional laboratory activity (bonus to laboratory grade)

## Passing the course

- Mandatory attendance to enter examination during 2022
- **L** grade  $\geq 5$  to enter examination during regular session
- **L**, **T** and **W** grades all  $\geq 5$  to pass the course

# Activity and grading

## Lecture 00

Lect. PhD.  
Arthur Molnar

Introduction  
to course  
Schedule  
Objectives  
Course content  
Bibliography  
Activity and  
grading

## Grading example

Suppose your grades are:

- Laboratory - 7
- Written - 7.50
- Practical - 6.80
- Seminar bonus - 0.30
- Laboratory bonus - 1

Your grade is calculated as:  $0.4 * (7 + 0.3 + 1) + 0.2 * 7.5 + 0.4 * 6.8 = 7.54$ , final grade is 8

# About the Practical Exam

Lecture 00

Lect. PhD.  
Arthur Molnar

Introduction  
to course  
Schedule  
Objectives  
Course content  
Bibliography  
Activity and  
grading

## About the Practical Exam

- Only **working** functionalities are graded
- Everything required for implementation will be studied
- Each problem will be interesting, in its own way
- Getting the extra points during the semester will help improve your grade

# Course Rules

## Lecture 00

Lect. PhD.  
Arthur Molnar

Introduction  
to course  
Schedule  
Objectives  
Course content  
Bibliography  
Activity and  
grading

- Seminar attendance mandatory **(10/14)**
- Laboratory attendance mandatory **(12/14)**
- Without making attendance you can't enter the exam this year!
- Detailed rules for laboratory activities are on the **General** channel, **Files** section
- **Be honest, solve the graded assignments by yourself, do not plagiarize!**