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

arthur@cs.ubbcluj.ro

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. Maria-Iuliana BOCICOR
- Lect. PhD. Mihoc Tudor
- Lect. PhD. Gaceanu Radu
- Assist. PhD. Mircea Ioan-Gabriel

Schedule

Lecture 00

Lect. PhD.
Arthur Molnar

Introduction
to course
Schedule
Objectives
Course content
Bibliography
Activity and
grading

- **Lectures**: 2 hours/week
- **Seminars**: 2 hours/week
- **Labs**: 2 hours/week

Course page:
http://www.cs.ubbcluj.ro/~arthur/Fundamentals of
Programming/

Email: **arthur@cs.ubbcluj.ro**.

Note!

We will only reply to emails from the @scs domain

Objectives

What should you gain from this course?

- Learn key programming concepts
- Learn basic concepts in software engineering (design, implementation and maintenance of software systems)
- Understand basic software tools (IDE's, debugger, testing tools)
- Learn the basics of the Python programming language
- Acquire and improve your programming style.

Course content

How is this course organized?

- Programming in the large
- Programming in the small

Course content

Lecture 00

Lect. PhD.
Arthur Molnar

Introduction
to course
Schedule
Objectives
Course content
Bibliography
Activity and
grading

- **Programming in the large**
    1. Introduction
    2. Procedural programming. Compound Types
    3. Test Driven Development
    4. Modular Programming
    5. Design guidelines in large scale programming
    6. Exceptions
    7. User Defined Types
    8. UML. Design Principles.
    9. Layered architecture
    10. Program Testing. Refactoring.

Course content

Lecture 00

Lect. PhD.
Arthur Molnar

Introduction
to course
Schedule
Objectives
Course content
Bibliography
Activity and
grading

- **Programming in the small**
    - 12 Recursion. Computational complexity
    - 13 Searching. Sorting
    - 14 Problem solving methods
- **Preparing for the exam**
    - 15 Overview

Bibliography

References

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

- 30% - Laboratory work (programs, documentations, tests **(L)**
- 30% - Practical test (in the regular session) **(T)**
- 40% - Written exam (in the regular session) **(W)**
- 0 - 0.5p Seminar activity (as bonus to final grade)
- 0 - 1p Laboratory activity (during the lab and at home, as bonus to final grade)

Minimum grade = **5 at L,T and W**

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 - 6.55
- Written - 7.50
- Practical - 6.80
- Seminar bonus - 0.40
- Laboratory bonus - 0.20

Your grade is calculated as: 0.3 * 6.55 + 0.4 * 7.5 + 0.3 * 6.8 + 0.4 + 0.2 = 7.00 + 0.4 + 0.2 = 7.60, final grade is 8)

Activity and grading

- Laboratory attendace mandatory (12/14)
- Seminar attendance mandatory (10/14)

NB!

Without making attendace you can't enter the exam this year!