



**BERGISCHE
UNIVERSITÄT
WUPPERTAL**

Prof. Dr.-Ing. Johannes Backhaus
Prüfungsausschuss Druck- und Medientechnologie

Fakultät für Elektrotechnik, Informationstechnik und
Medientechnik

Telefon +49 (0)202 439 1032
E-Mail jbackhaus@uni-wuppertal.de

Datum 04.10.2018

Studiengang Master DMT

Zusätzliches Modulangebot im Wahlpflichtbereich

Veranstalter: Prof. Dr. B. Gipp, Lehrstuhl Digitale Medien (neu seit WS 2018/19)

Introduction to Computing for Non-Computer Science Majors

Workload: 4 SWS (2 Lecture / 2 Exercise), 6 ECTS

LEARNING OBJECTIVES

Course participants will gain an overview of basic data structures and algorithms to solve problems in computer science and become familiar with the Python programming language. Through hands-on assignments, students will get an opportunity to gain a deeper understanding of the topics addressed in the lecture.

Through lectures, exercises and individual work, students will train their ability to:

- analyze a given problem from a computing point of view;
- choose appropriate data structures and algorithm for solving the problem;
- implement their solution using Python;
- deal with errors using online sources;
- prepare a documentation for the software developed;

By successfully completing the course, participants will acquire the knowledge and the skills required to successfully complete various forms of computer-science-related projects.

COURSE CONTENT DESCRIPTION

- **Theoretical Foundations**
- What is an algorithm
- Complexity of algorithms
- Mathematical logic
- Set theory
- Data types and encoding

Linear Data Structures

- Queue and stack
- Array and list
- Set and hash
- Searching and sorting in linear structures

Trees

- General trees
- Binary trees
- Searching and sorting in trees
- Modifications of binary trees

Graphs

- Terminology and properties
- Searching
- Finding optimal paths
- Finding minimum spanning trees
- Optimal matching

Python Programming Language

- Introduction, keywords, flow control
- Data types and structures
- Implementation of the algorithms mentioned above

TEACHING METHODS

- Following methods constitute core of the teaching:
- Interactive lectures to acquire theoretical knowledge
- Hands-on exercises, in which students solve applied problems to learn essential skills
- Individual assignments to train practical skills

LITERATURE

- Steven S. Skiena: The algorithm design manual. New York: Springer, 1998. ISBN 0387948600.
- Thomas H. Cormen, Charles E. Leiserson a Ronald L. Rivest. Introduction to algorithms. Cambridge: MIT Press, 1989. ISBN 0070131430.
- Thomas Foster: Logic, Induction and Sets. Cambridge University Press, 2003. ISBN 0-521-82621-7
- Al Sweigart: Automate the Boring Stuff with Python. Available online from <https://automatetheboringstuff.com/>
- John Zelle: Python Programming: An Introduction to Computer Science. Franklin, Beedle & Associates Inc. (2003). ISBN 978-1887902991
- Peter Wentworth, Jeffrey Elkner, Allen B. Downey, and Chris Meyers: How to Think Like a Computer Scientist: Learning with Python 3. Available online from <http://openbookproject.net/thinkcs/python/english3e/>
- Brad Miller and Runestone contributors: Interactive Python: http://interactivepython.org/runestone/default/user/login?_next=/runestone/default/index
- Python documentation: <https://www.python.org/doc/>

ASSESSMENT OF LEARNING

To successfully complete the course, students will be required to

- Submit solutions for assignments (50% of final grade)
- Pass a written test (50% of final grade)

Completion of all the deliverables is mandatory. Each of them will be evaluated separately, the overall grade will be calculated based on the weights of particular deliverables.

Anmeldung bis 12.10.2018 bei Frau S. Rosalen (Tel. 1308 srosalen@uni-wuppertal.de)
Termine der Lehrveranstaltung werden mit den interessierten Studierenden abgestimmt.