SOFTWARE ENGINEERING

MS in Earth Science Engineering / Specialisation in Geoinformatics Engineering

Semester 2, 2018/19

COURSE COMMUNICATION FOLDER

University of Miskolc
Faculty of Earth Science and Engineering
Institute of Geophysics and Geoinformatics

Course datasheet

<table>
<tr><td>Course Title: Software engineering

Instructors: Péter Tamás Vass Dr., associate professor</td><td>Code:
Responsible department/institute: Institute of Geophysics and Geoinformatics / Department of Geophysics
Type of course: Compulsory</td></tr>
<tr><td>Position in curriculum (which semester): 2</td><td>Pre-requisites (if any) MFGGT710003</td></tr>
<tr><td>No. of contact hours per week (lecture + seminar): 1+1</td><td>Type of Assessment (examination/ practical mark / other): examination</td></tr>
<tr><td>Credits: 3</td><td>Course: full time</td></tr>
<tr><td colspan="2">Course Description:
The main objective of the subject is to familiarize the students with the elements of programming necessary to solve engineering and scientific problems. The course tries to provide fundamental knowledge of programming in Python.
The short curriculum of the subject:
Introduction. Some basic concepts and definitions. The elements, design and descriptions of algorithms. Main features of programming languages. Fundamental steps of program development. A brief introduction to Python programming language, Anaconda distribution, and Spyder development environment. The use of Python in interactive and script modes. Python variable types and their operators. Python numbers and number type conversions. Built-in mathematical functions and constants. Built-in random number functions. Control structures in Python. Creating and calling user-defined functions. Managing files and directories. Text file input / output. Plotting graphs by means of the Matplotlib library. Technical and scientific computations by using the NymPy and ScyPy packages. Introduction to building up graphical user interfaces (GUI). GIS programming by using Python.
Competencies to evolve:
Knowledge: T1, T2, T4, T5, T6, T7, T9
Ability: K6, K8
Attitude: A1, A2, A3, A4, A5, A7
Autonomy and responsibility: F1, F2, F3, F4, F5</td></tr>
<tr><td colspan="2">Assessment and grading:
Condition for obtaining the signature: the presence in at least 60 % of the lessons.
The determination of the examination grade is entirely based on the result of examination.
Grading scale (% value → grade): 0 – 49 % → 1 (fail), 50 – 64 % → 2 (pass),
65 – 79 % → 3 (satisfactory), 80 – 89 % → 4 (good), 90 – 100 % → 5 (excellent).</td></tr>
<tr><td colspan="2">Compulsory or recommended literature resources:
The presented slides converted in pdf format: http://geofizika.uni-miskolc.hu/segedlet.html
Mark Summerfield, 2009: Programming in Python 3, Developer’s Library, ISBN: 0137129297
Hans Petter Langtangen 2008: Python Scripting for Computational Science, Third Edition, Springer-Verlag Berlin Heidelberg, ISBN 978-3-540-73915-9
The Python Tutorial: https://docs.python.org/3/tutorial/index.html</td></tr>
</table>

Syllabus of the semester

Week	Lecture and seminar
1	Introduction. Some basic concepts and definitions: computer program, software, algorithms, constants, variables, attributes of the variables. Elements of the algorithms: read, write, assignment, conditional branch, loop. The ways of expressing algorithms: natural languages, pseudocodes, flowchart.
2	Designing and expressing algorithms. Introduction to the use of a free flowchart editor. Doing exercises.
3	General characteristics of high-level programming languages. Semantics, syntax, compiler and interpreter. The fundamental steps of program development.
4	General characteristics of interpreted programming languages and their advantages. A brief introduction to Python programming language, Anaconda distribution, and Spyder development environment. The use of Python in interactive and script modes.
5	Python variable types and their operators. Examples.
6	Python numbers and number type conversions. Built-in mathematical functions and constants. Built-in random number functions. Examples.
7	Control structures in Python. Examples.
8	Creating and calling user-defined functions. Examples.
9	Managing files and directories. Text file input / output. Examples.
10	Plotting graphs by means of the Matplotlib library. Examples.
11	Technical and scientific computations by using the NymPy and ScyPy packages. Examples.
12	Introduction to building up graphical user interfaces (GUI). Examples.
13	GIS programming by using Python. Examples.
14	GIS programming by using Python.Summary.