|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **1. Course title:** Elementary programming | | | | |
|  | | | | |
| **2. Code:** | | **3. Type (lecture, practice etc.):** seminar | | |
|  | | | | |
| **4. Contact hours:** 4 hoursper week | | **5. Number of credits (ECTS):** 4 | | |
|  | | | | |
| 6. Preliminary conditions (max. 3): | | | | |
|  | | | | |
| **7. Announced:** fall semester, spring semester, both | | | | |
|  | | | | |
| **8. Limit for participants:** 150 | | | | |
|  | | | | |
| **10. Responsible teacher (faculty, institute and department):**  Dr. Mátyás Koniorczyk (Faculty of Science, Institute of Mathematics and Informatics, Department of Applied Mathematics) | | | | |
|  | | | | |
| 11. Teacher(s) and percentage: | | Ildikó Jenák | | 60 % |
|  | |  |
|  | |  |
|  | |  |
|  | |  |
|  | | | | |
| **12. Language:** English | | | | |
|  | | | | |
| 13. Course objectives and/or learning outcomes:  Objectives:  The aim of the course is to obtain a very basic insight into the basics of programming in a procedural language in an interpreter environment. The course uses the Python language. The main intention is to equalize the high-school level backround of entering students by establishing a common minimal knowledge and skill set expected from the students.  Learning outcomes: students completing the course will  *have a knowledge* on the most basic concepts of imperative-language programming such as, e.g. assignment, variable, control statements (conditionals, loops), arrays, etc. They use the suitable professional vocabulary of the topic.  They will be *able* to use the known concepts for implementing simple computer programs and solve basic programming problems.  They will be *open* and *intend* to significantly increase their programming abilities and skills.  They will be *able* *in a stand-alone way* to create simple programs in Python. | | | | |
|  | | | | |
| 14. Course outline   1. Installing and using Python. 2. Numbers, strings, variables. Assignments. Basic arithmetics. 3. Types, type conversions. Standard I/O. File I/O. 4. Basic GUI-s, simple windows. 5. Comparisons. Conditionals. 6. Loops. 7. Lists. 8. Functions. 9. Test. 10. Modules. Simple classes and objects. 11. Project work, additional topics according to current interests. 12. Project work, additional topics according to current interests. 13. Project work, additional topics according to current interests. | | | | |
|  | | | | |
| **15. Mid-semester works**  The lessons end with problems to be finished at home. Besides, each students is given a more complex task on the 10th week, which has to be solved, documented, and submitted. | | | | |
|  | | | | |
| **16. Course requirements and grading**  The semester concludes in a mark which is given based on the results of pre-announced short, practical tests, a more complex test on the 9th week and the stand-alone project work mentioned before. Tests concluding in mark 1 (insufficient) have to be repeated during the semester. | | | | |
|  | | | | |
| 17. List of readings   1. Warren D. Sande – Carter Sande: *Hello World! Computer Programming for Kids and Other Beginners*. Manning Publications Co., Greenwich (CT, USA), 2009. ISBN: 978-1-933988-49-8. | | | | |
|  | | | | |
| 18. Recommended texts, further readings   1. The actual documentation of Python: [www.python.org](http://www.python.org) 2. Python online tutorial: [www.learnpython.org](http://www.learnpython.org) | | | | |
|  | | | | |
| **Date** | 13 April, 2017 | **Prepared by** |  | |
| Dr. Mátyás KONIORCZYK  responsible teacher | |
|  | | | | |
| **Endorsed by** | | |  | |
| XXX program supervisor | |