|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1. Course title: Programming I. | | | | | |
|  | | | | |
| 2. Code: | | 3. Type (lecture, practice etc.): practice | | | |
|  | | | | |
| 4. Contact hours: 4 hoursper week | | 5. Number of credits (ECTS): 5 | | | |
|  | | | | |
| 6. Preliminary conditions (max. 3): | | | | | |
|  | | | | |
| 7. Announced:fall semester, spring semester, both | | | | | |
|  | | | | |
| 8. Limit for participants: 48 | | | | | |
|  | | | | |
| 10. Responsible teacher (faculty, institute and department):  Gimesi László PhD (Faculty of Science, Institute of Mathematics and Informatics) | | | | | |
|  | | | | |
| 11. Teacher(s) and percentage: | | Makkai Géza PhD | | 100 % | |
|  | |  | |
|  | |  | |
|  | |  | |
|  | |  | |
|  | | | | |
| 12. Language:English | | | | | |
|  | | | | |
| 13. Course objectives and/or learning outcomes:  Students will know basic programming structures, software development methodology, and more important programming environments. They will be introduced to C++ programming language and the basics of programming.  With the help of acquired methods students lay down the foundation for their further studies in programming. They will be able to practice algorithmic thinking, programming basic algorithms, and designing, coding, testing and managing individual programming tasks.  The bases for next semester’s Programming II. are developed in this course. | | | | | |
|  | | | | |
| 14. Course outline   1. Annunciation of course requirements. Foundations of algorithmic thinking. 2. Examples for aiding algorithmic thinking 3. Methods of program design. C++ programming environment. Loading the program, learning to use it, managing the framework. 4. Basic programming, basic syntax. Declaring integer and float variables, data representations, number constants, arithmetic operators, expressions. 5. Character type variables, character constants. Operators, precedencies. 6. Control structures: conditional control transfer (if, switch-case) 7. Debugging. 8. Control structures: iterations. 9. Pointers, arrays. 10. String variables. 11. Functions and its parameters. 12. Functions, standard functions. 13. Summary, evaluation of course fulfillment. | | | | | |
|  | | | | |
| 15. Mid-semester works   1. Assessment. 2. Test. 3. Test. 4. Test. 5. Test. 6. Test. 7. Assessment: individual project – development of a registration software | | | | | |
|  | | | | |
| 16. Course requirements and grading  The final grade will consist of the average of the followings:   * Week 3 assessment: 20% * tests: 80%, * Week 13 assessment: 0% (no grade, but compulsory for fulfilling the course)   Replacement or correctional tests are possible on the last week, or the first week of the exam period. | | | | | |
|  | | | | |
| 17. List of readings   1. C++ Language – Tutorials: http://www.cplusplus.com/doc/tutorial/ 2. Published tasks in [Neptun Meet Street](javascript:__doPostBack('upChooser$btnKollab','')) | | | | | |
|  | | | | |
| 18. Recommended texts, further readings   1. Any Internet publication about C++ | | | | | |
|  | | | | |
| **Date** | 24 April, 2017 | **Prepared by** |  | | |
| Dr. Gimesi László  Responsible teacher | | |
|  | | | | |
| **Endorsed by** | | |  | | |
| László Tóth, PhD  program supervisor | | |