|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1. Course title: Introduction to Computer Science | | | | | |
|  | | | | |
| 2. Code: | | 3. Type (lecture, practice etc.): practice | | | |
|  | | | | |
| 4. Contact hours: 4 hoursper week | | 5. Number of credits (ECTS): 7 | | | |
|  | | | | |
| 6. Preliminary conditions (max. 3): | | | | | |
|  | | | | |
| 7. Announced:fall semester, spring semester, both | | | | | |
|  | | | | |
| 8. Limit for participants: 150 | | | | | |
|  | | | | |
| 10. Responsible teacher (faculty, institute and department):  Sándor JENEI, DSc (Faculty of Science, Institute of Mathematics and Informatics, Department of Informatics) | | | | | |
|  | | | | |
| 11. Teacher(s) and percentage: | | Ildikó JENÁK | | 100 % | |
|  | |  | |
|  | |  | |
|  | |  | |
|  | |  | |
|  | | | | |
| 12. Language:English | | | | | |
|  | | | | |
| 13. Course objectives and/or learning outcomes  Objectives: The course intends to introduce students to basic concepts, results and methodology of the theory of computation. The classes will provide insight into non-trivial algorithms (with their corresponding pseudocodes), proofs of their correctness, and estimation procedures to determine complexity.  Learning outcomes: Students completing the course will have *knowledge* of basic notions, results and vocabulary of computing theory, and of pseudocode of basic algorithms. They will be *able* to estimate complexity of algorithms based on pseudocodes. They will have a *competence* of proving the correctness of algorithms. Their positive *attitude* towards mathematical problem solving will increase, furthermore they will be open to acquire algorithmic thinking and new methods of problem solving. | | | | | |
|  | | | | |
| 14. Course outline   1. Algorithmic problems, modelling. Running time of algorithms. 2. Insertion and sorting. 3. Packing and covering. Matching problem. Binary trees, decision trees. Graph search. Preorder, inorder and postorder tree traversals. 4. Connected graphs, shortest path. Knapsack problem. Suboptimal algorithms. 5. Shortest path on weighed graphs, Dijkstra’s algorithm. 6. Longest path, Eulerian and Hamiltonian graphs. Graph colouring, chromatic number, planar graphs, 4-colour theorem, Kuratowski’s and Wagner’s theorems. 7. Graph diagnostics. Tournament and its winner. Generalisation: logical formulas. 8. Polynomial division, Euclidean algorithm for integers and polynomials. Chinese Remainder Theorem. Sturm’s theorem. Evaluating polynomials. Horner-method. 9. Midterm exam (written). 10. Single-tape Turing machines. 11. Multi-tape Turing machines (2-tape and 3-tape). 12. Multi-tape Turing machines (k-tape). 13. Final exam (written). | | | | | |
|  | | | | |
| 15. Mid-semester works  Doing homework and is highly recommended. | | | | | |
|  | | | | |
| 16. Course requirements and grading  Attending classes is mandatory. Missing classes is allowed twice. One additional missing is accepted if an appropriate and authentic medical certificate justifies the cause of absence.  Written exams are based on lectures, accessible electronic sources, and course material.  The final course grade is determined by the arithmetic mean of the exam results (in percentages).  Final grades:   * 0–49%: fail * 50% - 61%: acceptable * 62% - 73%: average * 74% - 85%: good * 86% - 100%: excellent | | | | | |
|  | | | | |
| 17. List of compulsory readings   1. Gács, P.: Complexity of Algorithms, 1999. http://math.bme.hu/~gabor/oktatas/SztoM/GacsLov.complexity.pdf | | | | | |
|  | | | | |
| 18. Recommended texts, further readings  Selected parts from   1. Farlow, S. J.: Directed Graphs: Tournament Graphs. http://www.math.umaine.edu/~farlow/sec35.pdf 2. Harju, T.: Lecture Notes on Graph Theory, 2011. http://cs.bme.hu/fcs/graphtheory.pdf 3. Jia, Y. B.: Roots of Polynomials, 2015. http://web.cs.iastate.edu/~cs577/handouts/polyroots.pdf 4. Klappenecker, A.: Euclid’s Algorithm. http://faculty.cs.tamu.edu/klappi/alg/euclid.pdf 5. LaValle, S. M.: Planning Algorithms, Cambridge University Press, 2006. http://planning.cs.uiuc.edu/booka4.pdf 6. Mathcentre: Polynomial division, 2009. http://www.mathcentre.ac.uk/resources/uploaded/mc-ty-polydiv-2009-1.pdf 7. Schrijver, A.: A Course in Combinatorial Optimization, 2013. http://homepages.cwi.nl/~lex/files/dict.pdf 8. Tsishchanka, K.: Dividing Polynomials. https://cims.nyu.edu/~kiryl/Precalculus/Section\_3.3-Dividing%20Polynomials/Dividing%20Polynomials.pdf 9. Ullman, J. D.: Foundations of Computer Science, 1992. http://infolab.stanford.edu/~ullman/focs.html 10. Ward, K.: Horner’s Method or Scheme. http://www.trans4mind.com/personal\_development/mathematics/polynomials/hornerMethod.htm 11. Wirth, N: Algorithms and Data Structures, 1985. http://www.ethoberon.ethz.ch/WirthPubl/AD.pdf | | | | | |
|  | | | | |
| **Date** | 27 April, 2017 | **Prepared by** |  | | |
| Sándor JENEI, DSc  responsible teacher | | |
|  | | | | |
| **Endorsed by** | | | Dr. Mátyás KONIORCZYK program supervisor | | |
|  | | |