2. Course Title: The methodology of programming I.

3. Code:	**4. Kind (lecure, lab stb.): lecture**+lab
5. Nr. of hours weekly: 4 (2 lect.+2 lab)	**5**

/7. Preconditions (max. 3 preceeding course):

–

8. How often is it offered: ☐Fall semester, ☒Spring semester, ☐Both semesters

9. Maximal nr. of students: 24 per groups

10. Lecturer responsible: (faculty, institute and department):

Dr. Kilián Imre (Fac. of Sciences, Institute of Matematics and Informatics, Department of Information Technology and Biorobotics)

11. Lecturers with their procentual rates:	Dr. Kilián Imre	100%

12. Course language: English

13. Learning outcomes:

Main purpose of the course is that students own the methodology and usual approaches of programming and algorithm constructions, based primarily on C/C++ as a programming language.
Students, completing the course:
know the basic elements of C programing language, they *own* the basic approach for algorithm design,
able to express and create algorithms int he basic imperative approach, using C as a programming language
able to choose and to decide the methodology to be used for a given problem, and they are also able to apply it for the problem, and finally to implement the algorithm in a computer

14. Course program divided to 13 weeks:

1: Grouping of programming languages according to abstraction level, paradigm and programming subject
2: Chomsky's hierarchy of languages. CFG and RG languages and their appearance in the lexical and grammatical level of programming language
3: Control structures, assignment, sequence and block. Basic I/O instructions.
4: Expressions and their evaluation. Operators and their precedence.
5: Operations on linear data-structures. Summarizing, searching for minimal/maximal values, existencial and universal decision
6: Scope and lifetime of variables. Global and local variables. Iterated, union and direct-product datatype constructions.
7: Static and dynamic memory. Pointer and reference types. Typical memory management mistakes.
8: Subprograms and their implementations. Macroes. Parameter passing. Recursion. Handling exceptions.
9: Top-down and bottom-up algorithm design.
10: Problem space definitions. Algorithms working on principle of graph search.
11: Structured programming, program design and verification.
12: Program efficiency and its measuring.
13: Template types and their implementation

15. Special tasks during the semester:

16. Description of evaluation:

- oral exam

17. Required reading:

[1] Thomas H. Cormen, Charles E. Leiserson,Ronald L. Rivest, Clifford Stein: Introduction to Algorithms. MIT Press, 1990.

18. Proposed reading:

[1] O.-J. Dahl, Edsger W. Dijkstra, C. A. R. Hoare: *Structured Programming* (Academic Press, London, 1972)

This course description has been made:	15-March-2017.	**Created by:**	Dr. Kilián Imre lecturer
		Approved by:	