(*faculty stamp*)

COURSE DESCRIPTION

1. Course title: PROGRAM SPECIALIZATION AND METAPOGRAMMING	**2. Course code: WM2**
3. Validity of course description: 2020/2021	
4. Level of studies: 1st cycle of higher education	
5. Mode of studies: intramural studies	
6. Field of study: INFORMATICS	(FACULTY SYMBOL) RMS
7. Profile of studies: general	
8. Programme: all	
9. Semester: VI	
10. Faculty teaching the course: Faculty of Applied Mathematics	
11. Course instructor: professor Robertas Damaševičius	
12. Course classification: approved programme elective (monographic lecture)	
13. Course status: elective	
14. Language of instruction: English	
15. Pre-requisite qualifications: Programming, English.	
16. Course objectives: to acquire knowledge in metaprogramming and specialization techniques and acquire abilities in developing generic (reusable) and specialized software components.	

17. Description of learning outcomes:

A student who completes the course successfully should

Nr	Learning outcomes description	Method of assessment	Teaching methods	Learning outcomes reference code
1.	Is able to understand and explain the principles of metaprogramming and generative programming	project	lecture, laboratory	K2A_W04 K2A_U13
2.	Is able to understand and apply the main elements of feature modelling to develop generic and reusable software components (software generators)	project	lecture, laboratory	K2A_W04 K2A_W06 K2A_U13 K2A_U14
3.	Is able to understand and explain the concepts of program and data specialization	project	lecture, laboratory	K2A_W07 K2A_U13
4.	Is able to use program specialization techniques in practice to improve the performance of software components (functions)	project	lecture, laboratory	K2A_U13 K2A_W05 K2A_W06
5.	Is able to use the elements of C++ language (C++ templates) to develop generic C++ classes and functions	project	lecture, laboratory	K2A_W13 K2A_K01 K2A_K06

18. Teaching modes and hours

Lecture / BA /MA Seminar / Class / Project / Laboratory

Lecture 30h. Laboratory 30h.

19. Syllabus description:

Lecture: Fundamental concepts of meta-programming. Background of meta-programming techniques. Homogenous and heterogenous metaprogramming. Generative metaprogramming. Template metaprogramming. Metaprogram modelling using Feature Diagrams. Program specialization and data specialization. Improvement of program performance using program specialization.

Laboratories: Practical approach to theory and examples presented at lectures.

20. Examination: no

21. Primary sources:

1. V. Štuikys and R. Damaševičius. Meta-Programming and Model-Driven Meta-Program Development (Principles, Processes and Techniques), 2012, Springer.
2. Renaud Marlet. Program specialization. Wiley, 2013.
3. Krysztof Czarnecki, Ulrich Eisenecker. Generative Programming: Methods, Tools, and Applications. 1st Edition. Addison-Wesley Professional; 1 edition (2000).

22. Secondary sources:

1. Štuikys V., Damaševičius R. (2013) Meta-Meta-Programming and Equivalent Transformations of Heterogeneous Meta-Programs. In: Meta-Programming and Model-Driven Meta-Program Development. Advanced Information and Knowledge Processing, vol 5. Springer, London.
2. D. Abrahams, A. Gurtovoy, C++ Template Metaprogramming: Concepts, Tools, and Techniques from Boost and Beyond (C++ in Depth) 2004

23. Total workload required to achieve learning outcomes

Lp.	Teaching mode :	Contact hours / Student workload hours
1	Lecture	30/10
2	Classes	/
3	Laboratory	30/20
4	Project	/30
5	BA/ MA Seminar	/
6	Other: consultations, use of e-learning webpage	/30
	Total number of hours	60/90

24. Total hours: 150

25. Number of ECTS credits: 5

26. Number of ECTS credits allocated for contact hours: 4

27. Number of ECTS credits allocated for in-practice hours (laboratory classes, projects): 1

26. Comments:

To pass, it is necessary to achieve all learning outcomes described above.

Approved:

…….
(date, Instructor's signature)

……
(*date , the Director of the Faculty Unit signature)*