MODULE DESCRIPTION

Module title	Module code
Procedural Programming	

Lecturer(s)	Department where the module is delivered
Coordinator: Viktoras Golubevas **Other lecturers:** assoc. prof. dr. Saulius Ragaišis	Department of Software Engineering Faculty of Mathematics and Informatics Vilnius University

Cycle	Type of the module
First	Compulsory

Mode of delivery	Semester or period when the module is delivered	Language of instruction
Face-to-face	1 semester	Lithuanian

Prerequisites
Prerequisites: Procedural programming, Object-oriented Programming

Number of credits allocated	Student‘s workload	Contact hours	Self-study hours
5	134	68	66

Purpose of the module: programme competences to be developed		
Purpose of the module – to provide knowledge and to gain skills of procedural programming using C programming language; to develop disciplined programming skills. ***Generic competences:*** • Communication and collaboration (GK1). • Life-long learning (GK2). • Social responsibility (GK3). ***Specific competences:*** • Knowledge and skills of underlying conceptual basis *(SK4).* • Technological and methodological knowledge and skills, professional competence *(SK6).*		
Learning outcomes of the module: students will	**Teaching and learning methods**	**Assessment methods**
Strengthen the knowledge and skills of procedural programming.	Problem-oriented teaching, self-study of literature, laboratory works, individual work.	Laboratory works, results presentation, written exam (open, semi-open and close-ended questions and tasks).
Understand effective "low" level pointer mechanism and its applications		
Perceive the principles of modular programming: interface, implementation, data abstraction, information hiding.		
Know the basic syntax elements of C language and will be able to easier master other C family programming languages (C++, C #, Java, PHP, and Perl).		
Understand the need of disciplined programming, have initial skills.	Problem-oriented teaching, self-study of literature, laboratory work	Laboratory works, results presentation.

Content: breakdown of the topics	**Contact hours**							**Self-study work: time and assignments**	
	Lectures	Tutorials	Seminars	Practice	Laboratory work (LW)	*Tutorial during LW*	**Contact hours**	**Self-study hours**	**Assignments**
History of C programming language, comparative characteristic.	2				2	*8*	**4**	**4**	Self-study of literature, laboratory works, individual work.
Structure of C program, functions (comparison of simple C and Pascal programs).	2				2		**4**	**4**	
Language basics (data types, statements, operators).	4				4		**8**	**8**	
The C system (preprocessor, compiler, standard library).	2				2		**4**	**4**	
An overview of standard library.	2				2		**4**	**4**	
Personal Software Process: introduction, basic principles, main metrics, coding standard, reviews.	2				2		**4**	**6**	
Pointers, pointer arithmetic.	3				3		**6**	**6**	
Pointers and arrays, pointers and constants.	3				3		**6**	**6**	
Parameters passing (call by value, call by reference).	2				2		**4**	**4**	
Types of memory, dynamic memory allocation.	2				2		**4**	**4**	
User defined data types (typdef, enum, struct, union).	2				2		**4**	**4**	
Data structures, its allocation in dynamic memory.	2				2		**4**	**4**	
Storage class specifiers, information hiding.	2				2		**4**	**4**	
Principals of modular programming in C.	2				2		**4**	**4**	
Preparation for the exam and taking the final exam (written).		2					**4**		2 hours for preparation, 2 hours for exam
Total	**32**	**2**			**32**	*8*	**68**	**66**	

Assessment strategy	**Weight %**	**Deadline**	**Assessment criteria**
Laboratory work No. 1	5	Week 5	Simple algorithm using basic language elements (conditions, loops). Work is assessed in 10 points system. Lateness leads to the decrease of the maximal assessment by 20% for every delayed week. Points multiplied by 0.05 are added to the final assessment.
Laboratory work No. 2	10	Week 8	More complex algorithm using arrays, pointers, different parameters passing techniques. Coding standard and compliance of program to the standard. Work is assessed in 10 points system. Lateness leads to the decrease of the maximal assessment by 20% for every delayed week. Points multiplied by 0.1 are added to the final assessment.
Laboratory work No. 3	15	Week 11	Allocation of dynamic memory, text analysis, files processing. Work is assessed in 10 points system. Lateness leads to the decrease of the maximal assessment by 20% for every delayed week. Points multiplied by 0.15 are added to the final assessment.

Laboratory work No. 4	20	Week 14	User defined data types, multi-file program. Work is assessed in 10 points system. Lateness leads to the decrease of the maximal assessment by 20% for every delayed week. Points multiplied by 0.2 are added to the final assessment.
Exam in written form	50	Exam session	A student can take part in the examination only if he/she has collected at least 2 points from laboratory works. The exam consists of 10-15 close-ended questions and 2-3 open and/or semi-open questions and tasks. Questions and tasks are formulated from topics set out in lectures.

Author	**Publishing year**	**Title**	**Number or volume**	**Publisher or URL**
Required reading				
B.W.Kernighan, D.M.Ritchie,	1988	The C Programming Language, Second edition		Prentice Hall
S. Ragaišis	2007	Personal Software Process (in Lithuanian)		http://www.mif.vu.lt/~ragaisis/PSP2007/Asmeninis.programu.kurimo.procesas.pdf
Recommended reading				
N. Parlante	2003	Essential C		http://cslibrary.stanford.edu/101/
A. Mehta	1995	A Crash Course in C		http://www.mattababy.org/~belmonte/Teaching/CCC/handouts.pdf
W.S. Humphrey	1997	Introduction to the Personal Software Process		Addison-Wesley