COURSE UNIT DESCRIPTION

Course unit title	Code
Procedural programming	

Annotation
The study subject is intended for the students of the study program Software engineering. During the studies, the stages of program development, the basic constructions of the programming language and data structures are introduced. Learn to create and combine structured and multi-file applications.

Lecturer(s)	Department, Faculty
Coordinating: prof. dr. Sigita Turskienė **Other: Lect. Dr. Dainius Balbonas**	Šiauliai Academy

Study cycle	Type of the course unit
First cycle studies	Compulsory

Mode of delivery	Semester or period when it is delivered	Language of instruction
Face-to-face	1 semester	Lithuanian/English

Requisites	
Prerequisites: No	**Co-requisites (if relevant):** basics of discrete mathematics

Number of ECTS credits allocated	Student's workload (total)	Contact hours	Individual work
5	133	64	69

Purpose of the course unit: programme competences to be developed		
The subject of procedural programming studies aims to develop analytical thinking by acquiring knowledge of procedural programming and initial skills to independently create programs that develop disciplined programming skills.		
Learning outcomes of the course unit	**Teaching and learning methods**	**Assessment methods**
Will be able to analyze and systematize the data required to implement frequently used algorithms.	Laboratory works, Literature analysis, Practical tasks, Lectures	Exam (written), Individual homework, Laboratory work defense
Will be able to apply procedural programming knowledge in various areas of software development	Laboratory works, Practical tasks, Lectures, Problem-based learning	Exam (written), Individual homework, Laboratory work defense
Will know the fundamental syntax elements of the C family languages, will be able to learn easily new programming	Laboratory works, Practical tasks, Lectures, Problem-based learning	Exam (written), Individual homework, Laboratory work defense

languages (eg C #, Java, Python, PHP))		
Will be able to create and debug various applications.	Laboratory works, Practical tasks, Lectures	Exam (written), Individual homework, Laboratory work defense
Will be able to plan the time, work independently, will acquire the basic skills of disciplined programming.	Laboratory works, Practical tasks	Exam (written), Individual homework, Laboratory work defense

Course content: breakdown of the topics	**Contact hours**							**Individual work: time and assignments**	
	Lectures	Tutorials	Seminars	Workshops	Laboratory work	Internship/work placement	**Contact hours, total**	**Individual work**	**Assignments**
1. Program development stages. The concept of procedural programming. Programming environment. Preprocessor. Compiler.	3						**3**	**2**	Self-study of subject literature, exam (written)
2. Basic elements of programming language. Standard function libraries.	4				4		**8**	**2**	Self-study of subject literature, exam (written)
3. Major control and management operators. Conditional and variant sentences. Loop sentences. Iterative calculations.	4				6		**10**	**2**	Laboratory work No. 1, Self-study of subject literature, exam (written)
4. Functions and their parameters. Overlay functions. Recursive functions.	4				4		**8**	**4**	Laboratory work No. 3, Self-study of subject literature, exam (written)
5. Character string. Data files.	4				4		**8**	**4**	Laboratory work No. 5, Self-study of subject literature, exam (written)
6. User-defined data types. Arrays. Structures. Compounds.	4				4		**8**	**4**	Laboratory work No. 2, Self-study of subject literature, exam (written)
7. Dynamic memory management. Pointer types. Arithmetic of pointers. Arrays of pointers.	4				6		**10**	**4**	Laboratory work No. 4, Self-study of subject literature, exam (written)
8. Memory class specifiers. Visibility and existence of variables. Namespaces.	2				2		**4**	**2**	Homework No. 1, Self-study of subject literature, exam (written)
9. Creating multi-files programs. Structured files.	3				2		**5**	**2**	Homework No. 1, Self-study of subject literature, exam (written)
10. Exam.								**43**	Exam (3 h), for prepration (40 h)

Total	**32**				**32**		**64**	**69**	

Assessment strategy	Weight %	Deadline	Assessment criteria
Laboratory work 1	10	During semester	The student performs a laboratory work task from the loops and conditional sentences themes. The student must specify the problem formulated in the task. Laboratory work is evaluated from 0 to 10 points. Evaluation Criteria: • Implementation of the requirements specified in the task, accurate application of appropriate statements. (70%) • Ability to substantiate the implemented solutions, compare them with possible alternative solutions. (20%) • Evaluate the complexity and problems of the realized situation. (10%) • If laboratory work not performed (0 %)
Laboratory work 2	10	During semester	The student performs a laboratory task according to a more complex algorithm using arrays. The student must specify the problem formulated in the task. Laboratory work is evaluated from 0 to 10 points. Evaluation Criteria: • Implementation of the requirements specified in the task, accurate application of appropriate statements. (70%) • Ability to substantiate the implemented solutions, compare them with possible alternative solutions. (20%) • Evaluate the complexity and problems of the realized situation. (10%) • If laboratory work not performed (0 %)
Laboratory work 3	10	During semester	The student performs a laboratory task - creation of functions. The student must specify the problem formulated in the task. Laboratory work is evaluated from 0 to 10 points. Evaluation Criteria: • Implementation of the requirements specified in the task, accurate application of appropriate statements. (70%) • Ability to substantiate the implemented solutions, compare them with possible alternative solutions. (20%) • Evaluate the complexity and problems of the realized situation. (10%) • If laboratory work not performed (0 %)
Laboratory work 4	10	During semester	The student performs a laboratory task using dynamic memory and pointers. The student must specify the problem formulated in the task. Laboratory work is evaluated from 0 to 10 points. Evaluation Criteria: • Implementation of the requirements specified in the task, accurate application of appropriate statements. (70%) • Ability to substantiate the implemented solutions, compare them with possible alternative solutions. (20%) • Evaluate the complexity and problems of the realized situation. (10%) • If laboratory work not performed (0 %)
Laboratory work 5	10	During semester	The student performs a laboratory task using character sequences. The student must specify the problem formulated in the task. Laboratory work is evaluated from 0 to 10 points. Evaluation Criteria: • Implementation of the requirements specified in the task, accurate application of appropriate statements. (70%) • Ability to substantiate the implemented solutions, compare them with possible alternative solutions. (20%) • Evaluate the complexity and problems of the realized situation. (10%) • If laboratory work not performed (0 %)
Homework 1	10	During semester	The student individually performs a homework assignment from the development of multi-file programs The student must specify the problem formulated in the task. Homework is evaluated from 0 to 10 points. Evaluation Criteria:

			• Implementation of the requirements specified in the task, accurate application of appropriate statements. (70%) • Ability to substantiate the implemented solutions, compare them with possible alternative solutions. (20%) • Evaluate the complexity and problems of the realized situation. (10%) • If laboratory work not performed (0 %)
Exam	40	During exam session	The exam covers the knowledge and skills developed in all course topics. The exam test consists of 10 open questions and / or tasks. Each question tests the understanding of the concepts, data types or language constructions examined, the application of the material in an imaginary practical situation. Each task is evaluated with 1 point. Final evaluation: • Excellent and very good subject knowledge and skills. (10-9 correct answers) • Good knowledge and skills, there may be minor mistakes. (8 correct answers) • Average knowledge and skills, there are mistakes. (7 correct answers) • Knowledge and skills are below average, there are significant mistakes. (6 correct answers) • Knowledge and skills still meet the minimum requirements. Lots of mistakes. Level of knowledge and understanding. (5 correct answers). • Minimum requirements are not met. 0-4 correct answers.

Author	Publishing year	Title	Issue of a periodical or volume of a publication; pages	Publishing house or internet site
Required reading				
Brian W. Kernighan, Dennis M. Ritchie	**2015**	**The C Programming Language**		**Pearson Prentice Hall**
S. Qualline	**2010**	**Practical C++ programming**		**O'Reilly,**
P. Deitel, H. Deitel	**2014**	**C++ how to program**		**Pearson Prentice Hall**
Recommended reading				
C. Hortrsman	**2009**	**C++ for everyone**		**Wiley**
Andrew Koenig	**2006**	**Accelerated C++: Practical Programming by Example**		**Addison-Wesley**
E. Bruce, A. Chuck	**2004**	**Thinking in C++ Vol.2**		**Pearson Prentice Hall**
		Tutorials in internet		https://www.cpluspl us.com/ https://www.w3sch ools.com/CPP/defa ult.asp https://www.learncp p.com/ https://www.geeksf orgeeks.org/c-plus-plus/