COURSE UNIT DESCRIPTION

Course unit title	Code
Programamming Languages	

Annotation
Students get familiar with the development trends of programming languages, formal grammars, basic constructions of programming language and data structures. They learn to evaluate and to select the programming language in various aspects for particalar solutions.

Lecturer(s)	Department, Faculty
Coordinating: dr. Vaidas Giedrimas	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	Spring semester	Lithuanian/English

Requisites	
Prerequisites:	**Co-requisites (if relevant):**

Number of ECTS credits allocated	Student's workload (total)	Contact hours	Individual work
5	133	64	71

Purpose of the course unit: programme competences to be developed		
The subject of programming languages aims to develop analytical thinking, master the insights of programming languages development trends, syntax and semantics, formal grammar, criteria for assessing the quality of programming languages and practical skills to independently evaluate and select programming languages.		
Learning outcomes of the course unit	**Teaching and learning methods**	**Assessment methods**
Will be able to analyze the development trends of programming languages, types of classifications, implementation mechanisms.	Interactive lecture, analysis of references	Exam
Will know formal grammars and languages.	Interactive lecture, laboratory work, analysis of references	Exam, Defense of laboratory work
Will be able to perform a comparative analysis of the syntax and semantics of the main programming languages.	Interactive lecture, laboratory work, analysis of references	Exam, Defense of laboratory work, Homework
Know the evaluation criteria of programming languages.	Interactive lecture, Discussion, laboratory work, analysis of references	Exam, Defense of laboratory work.
Will be able to consistently explain programming paradigms and algorithms in typical and new areas of application systems projects.	Interactive lecture, laboratory work, analysis of references	Exam

Will be able to work independently, will acquire practical skills in selecting the appropriate programming languages.	Laboratory work, Discussion	Exam, Defense of laboratory work, Homework

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. The evolution of programming languages, its trends. Programming paradigms.	4				0		**4**	**5**	Independent study of subject literature, exam
2. Formal methods of language description. Vocabulary. Syntax. Semantics.Formal grammars and their taxonomy.	4				4		**8**	**5**	Independent study of subject literature, exam
3. Contextual languages. Regular languages. Free context languages. Unambiguity of grammars and languages.	4				4		**8**	**5**	Laboratory work 1,Independent study of subject literature, exam
4. BNF. Syntax diagrams.	2				4		**6**	**5**	Laboratory work 2, Independent study of subject literature, exam
5. Lexicon. Data types and structures.	4				6		**10**	**5**	Laboratory work 3,Independent study of subject literature, exam
6. Control structures	6				6		**12**	**10**	Laboratory work 4, Independent study of subject literature, exam
7. Procedures and functions	4				6		**10**	**10**	Homework, Independent study of subject literature, exam
8. Programming language standards. Translators. Compatibility of programming languages.Evaluation criteria for programming languages.	4				2		**6**	**9**	Independent study of subject literature, exam
9. Preparation for exam								**17**	
Total	**32**				**32**		**64**	**71**	

Assessment strategy	Weight %	Deadline	Assessment criteria
Laboratory work 1	10%	During the semester	The student individually performs a laboratory work task from the topic of language grammar. 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%). • Without laboratory work (0%)
Laboratory work 2	10%	During the semester	The student individually performs a laboratory work task from the topic of syntax diagrams. 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 solution (10%). • Without laboratory work (0%)
Laboratory work 3	10%	During the semester	The student individually performs a laboratory work task from the analysis of data types and structures of selected languages. 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 solution. (10%). • Without laboratory work (0%)
Laboratory work 4	10%	During the semester	The student individually performs a laboratory work task from the analysis of selected language management structures. 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 solution. (10%). • Without laboratory work (0%)
Homework	10%	During the semester	The student individually performs a homework assignment from an analysis of selected language features and procedures. 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 solution. (10%). • Undelivered homework (0%)
Exam	50%	Exam‘s session	The exam covers the knowledge and skills developed in all course topics. The exam test consists of 10 open-ended questions and / or tasks. Each question examines the application of the concepts and 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				
R.W. Sebesta	2012	Concepts of Programming Languages. 10 th ed. Pearson, 816 pages.		https://vulms.vu.edu.pk/Courses/CS508/Downloads/Concepts%20of%20Programming%20Languages%2011th%20Ed.pdf
Maurizio Gabbrielli, Simone Martini (2010).	2010	Programming Languages: Principles and Paradigms (Undergraduate Topics in Computer Science). 440 pages		Springer, http://websrv.dthu.edu.vn/attachments/newsevents/content2415/Programming_Languages_-_Principles_and_Paradigms_thereds1106.pdf
Arvind Kumar Bansal	2013	Introduction to Programming Languages, 624 pages.		Chapman and Hall/CRC
Recommended reading				
D. P. Friedman, M. Wand	2008	Essentials of Programming Languages, 3 ed., 432 pages.		MIT Press
Bjørner, D. Henson M.C	2008	Logics of Specification Languages		Springer,