

Subject code	ECTS credits
INF1003	6

Course title in Lithuanian

PROGRAMAVIMO PAGRINDAI

Course title in English

PROGRAMMING FUNDAMENTALS

Short course annotation in Lithuanian (up to 500 characters)

Dalyko tikslas – supažindinti studentus su programavimo principais, išmokyti C++ programavimo kalbos sintaksę, išaiškinti programų kūrimo etapus ir taisykles. Kurso metu studentai supažindinami su sąlyginiais ir ciklo sakiniais, išmokomi skaitinių ir tekstinių reikšmių tvarkymo, paieškos, skaičiavimų. Paieškos, įterpimo, išmetimo, rikiavimo algoritmų su vienmačiais ir dvimačiais masyvo elementais principų. Aiškinami programavimo stiliaus, kultūros bei programos dokumentavimo pagrindai. Išklause kursą studentai sugebės sudaryti taikomųjų uždavinių sprendimo algoritmus, gebės sukurti programas ir parengti dokumentaciją.

Short course annotation in English (up to 500 characters)

The course will cover basic terminology of structural programming, input/output control, decision control, repetition, subroutines, analyses of the data in one and two dimensional arrays, elementary strings and file processing. After finishing the course students will be able to develop algorithms and write computer instructions to solve problems, will learn programming in C language. Teaching methods are: lectures and laboratory works.

Prerequisites for entering the course

No prior programming experience required.

Course aim

Understand of designing of an algorithm and developing programm.

Links between course outcomes, criteria of learning achievement evaluation, study methods and methods of learning achievement assessment

No	Course outcomes	Criteria of learning achievement evaluation	Study methods	Methods of learning achievement assessment
1.	Knowledge and understanding of computer structure.	Student demonstrates the ability to explain processes going in computers.	Giving interpretations and illustrations through visual material; Practical works;	Evaluation of oral presentation and analysis of the practical works. Observations of students presentations, individual practical activities.
2.	Choose and apply software to solve practical problems	Ability to distinguish the programming languages and compound technologies. To show great consideration for inputs-outputs. Students presents the practical works to lecture and their colleagues.	Building a problem set. Giving interpretations and illustrations through visual material; Practical works;	Observations of students works, presentations, individual practical activities. Evaluation of semester work, written reports, classroom tests, written mid-term and final examinations.

3.	Provide knowledge on building of an algorithm, developing a program, providing an analysis of working, or nonworking program.	To show great consideration for inputs-outputs. Implement the search, insertion, removal and sorting algorithms. Manage one-dimensional and two-dimensional arrays and text file handling. Student presents the practical works to lecture and their colleagues.	Building a problem set. Giving interpretations and illustrations through visual material; Practical works; Reviewing material;	Observations of students works, presentations, individual practical activities. Evaluation of semester work, written reports, classroom tests, written mid-term and final examinations.
4.	Choose and apply suitable tools, interpret the results.	The ability to use received knowledge in other university courses. Student demonstrates skills in developing systems and applying the tools.	Giving interpretations and illustrations through visual material; Reviewing material;	Observations of students presentations, individual practical activities. Evaluation of oral presentation and analysis of the practical works, written reports, classroom tests, written mid-term and final examinations.

Links between study programme outcomes and course outcomes

Study programme outcomes	Running number of course outcome			
	1	2	3	4
Know and comprehend the needs and importance of information technologies in study process, also be able to apply programming knowledge and skills, data structures and modelling	+	+		
Identify the problem, collect and analyze real/theoretical data using various mathematical methods, tools and IT technologies			+	+

Content

No	Content (topics)
1.	Algorithms, data types and variables
2.	Handling of inputs-outputs and the flow control: loop and switch
3.	Functions and principles of computer programming
4.	Handling the text file data
5.	Working with the elements of one-dimensional array
6.	Analysis and calculations with elements of two-dimensional array
7.	Explaining the constructions of an algorithms for calculations (sum, average, number of positives, etc.), finding the minimum or the maximum, the sum of elements by criteria.
8.	Explaining the constructions of an algorithms for searching, sorting, interchanging.

Distribution of workload for students (contact and independent work hours)

Practicum	75 hours
Individual students work	85 hours
Total:	160 hours

Structure of cumulative score and value of its constituent parts

Final written exam (50%), mid-term written exam (17%), and assessments of laboratory (practical) work (33%).
--

Recommended reference materials

No.	Publication year	Authors of publication and title	Publishing house	Number of copies in		
				University library	Self-study rooms	Other libraries
Basic materials						
1.	2016	V.Barzdaitis „Programavim o pagrindai“- distance learning course				Electronic papers, in distance learnig system: http://moodle.vdu.lt
2.	2014	Programming video/text lectures				http://programavimopamokos.net / index.php?vaizdas=CPP
3.	2016	Aurimas Šimkus Programming tutorials				http://coderland.lt/
Supplementary materials						
1	2014	Ivor Horton. „Beginning Visual C++“	SlideShare	Free resources on SlideShare: http://www.slideshare.net/hodienloi/ivor-hortons-beginning-visual-c-2013		
2	2016	Visual Studio Quick Reference Guidance		Free resources on Internet: https://vsarquickguide.codeplex.com		
3	2016	Visual C++ Developer Center		Free resources on Internet: https://msdn.microsoft.com/en-us/vstudio/aa718325.aspx		
4	2016	CPP programming tutorials, best practice examples, working examples, debugging instructions		http://www.bogotobogo.com/cplusplus/cpptut.php http://www.cplusplus.com http://www.learncpp.com/		
5		Free forums resources: best news, issues solving solutions.		http://stackoverflow.com/questions/388242/the-definitive-c-book-guide-and-list https://www.quora.com/What-are-the-best-C++-books		

Course programme designed by

Lect. Vytautas Barzdaitis