



COURSE UNIT (MODULE) DESCRIPTION

Course unit (module) title	Code
Component and Parallel Oriented Programming	

Academic staff	Core academic unit(s)
Coordinating: dr. V. Giedrimas	Šiauliai Academy
Other:	

Study cycle	Type of the course unit
First cycle	Mandatory

Mode of delivery	Semester or period when it is delivered	Language of instruction
Auditorium	6th semester	Lithuanian/ English

Requisites	
Prerequisites: Procedural programming Object oriented programming	Co-requisites (if relevant):

Number of ECTS credits allocated	Student's workload (total)	Contact hours	Individual work
5	133	56	77

Purpose of the course unit
<p>To acquaint students with the paradigms of component and parallel programming, to reveal the influence of the repeated use of program units and the simultaneous execution of individual threads on the quality of program systems and the progress of the system development project.</p> <p>Cultivated competences:</p> <ul style="list-style-type: none"> BK1 Application of knowledge BK2 Social skills BK3 Personal abilities DK2 Abilities to conduct program system research <p>DK3 PS Special Abilities</p>

Learning outcomes of the course unit	Teaching and learning methods	Assessment methods
Will know the paradigm of component programming, features of component programming.	Interactive lecture	Exam
You will learn to create software components and global network services	Interactive lecture, laboratory works	Exam, Defense of laboratory work, Independent work.
You will learn to create components that use application systems.	Interactive lecture, laboratory works	Exam, Defense of laboratory work, Independent work.
Will know the basic principles of parallel programming, typical parallel algorithms.	Interactive lecture, laboratory works	Exam, defense of laboratory work
Will be able to create parallel processes and apply them in various areas.	Interactive lecture, laboratory works	Exam, defense of laboratory work

Content	Contact hours							Individual work: time and assignments	
	Lectures	Tutorials	Seminars	Workshops	Laboratory work	Internship	Contact hours, total	Individual work	Tasks for individual work
1. Component-based programming paradigm.	2			0	0		2	6	Exam
2. Life cycle of component programs. The concept of a component.	2				2		4	6	Exam, Defense of laboratory work, Independent work
3. Component models (.NET, EJB, OSGi, SCA)	4				8		12	10	
4. Heterogeneous component systems.	2				2		4	8	
5. Testing and support of component systems.	4				4		8	8	
6. Concept and types of parallel processes.	4				0		4	6	Exam
7. Multi-processor and multi-core architecture.	4				0		4	8	Exam, Defense of laboratory work, Independent work
8. Programming tools for parallel processes and features of their use.	4				2		6	8	
9. Parallel programming algorithms.					6		6	10	
10. Testing of parallel processes	2				4		6	7	
Total	28				28		56	77	

Assessment strategy	Weight %	Deadline	Assessment criteria
Defense of laboratory work	40%	Every second week	Laboratory works and their defense are evaluated. A total of 16 laboratory works
Presentation of independent work results	20%	2 times per semester	It is given in the first week of studies and is carried out in stages.
Exam	40%	During the exam session	The exam test in the Moodle environment consists of 20 open and closed type questions, each graded with half a point. The grade of the exam is equal to the sum of the collected points.

Author (-s)	Publishing year	Title	Issue of a periodical or volume of a publication	Publishing house or web link
Required reading				
Szyperski, C.	2002	Component software.		Addison-Wesley,
Whitehead, K.	2007	Component-based development principles and planning for business systems : principles and planning for business systems.		Addison-Wesley
McCool, M.	2012	Structured parallel programming patterns for efficient computation.		Elsevier, http://www.sciencedirect.com/science/book/9780124159938
Coulouris, G. et al.	2012	Distributed systems :concepts and design		Boston: Addison-Wesley

Recommended reading				
J. Löwy.	2005	Programming .NET Components		O'Reilly
Heineman, G. T. , Council, W. T.	2001	Component-based software engineering : putting the pieces together		Boston: Addison Wesley
Cheesman, J., Daniels. J.	2002	UML Components: A Simple Process for Specifying Component-Based Software		Addison-Wesley
Andy Ju An Wang. Kai Qian.	2006	Component-oriented programming.		Wiley
Crnkovic, I., Larsson M.	2002	Building Reliable Component-based Software Systems.		Artech House
Freeman, A.	2010	NET 4 Parallel Programming in C#.		Apress
Pacheco, P.	2011	An Introduction to Parallel Programming.		Morgan Kaufmann