



## COURSE UNIT DESCRIPTION

Course unit title	Course unit code
Software engineering I	

Lecturer(s)	Department where the course unit is delivered
<b>Coordinator:</b> Donatas Kimutis, Vytautas Ašeris <b>Other lecturers:</b> -	Department of Software Engineering Faculty of Mathematics and Informatics Vilnius University

Cycle	Type of the course unit
1 <sup>st</sup> (BA)	Compulsory

Mode of delivery	Semester or period when the course unit is delivered	Language of instruction
Face-to-face	3 <sup>rd</sup> semester	English, Lithuanian

Prerequisites
<b>Prerequisites:</b> Procedural programming, Object-oriented Programming

Number of credits allocated	Student's workload	Contact hours	Individual work
10	270	82	188

Purpose of the course unit: programme competences to be developed		
Purpose of the course unit – to get acquainted with software development methods using C # programming language and .NET framework, to consolidate knowledge of object-oriented programming.		
<b>Generic competences:</b> <ul style="list-style-type: none"> <li>• Communication and collaboration (GK1).</li> <li>• Life-long learning (GK2).</li> </ul>		
<b>Specific competences:</b> <ul style="list-style-type: none"> <li>• Knowledge and skills of underlying conceptual basis (SK4).</li> <li>• Software development knowledge and skills (SK5)</li> <li>• Technological and methodological knowledge and skills, professional competence (SK6).</li> </ul>		
Learning outcomes of the course unit: students will be able to	Teaching and learning methods	Assessment methods
Design, implement and develop applied programs, apply code reviews. Apply knowledge of software systems engineering, make qualified design and architectural decisions while expanding the functionality of the developed system. Combine theory and practice using .NET framework technologies and developing OO application systems. Develop the knowledge about data types, named and optional arguments as well as other new features of C# programming language. Program in C# independently and in a team, applying basic OO design templates using C# programming environment. Recognize the need for continuous learning and will have the initial skills. Work in the team - on site and remotely.	Lectures, problem-oriented teaching, team work, case studies, information retrieval, literature reading, individual work, learning from teammates, examples analysis, tutorials, laboratory works.	Laboratory works, results presentation, written exam (open, semi-open and close-ended questions and tasks).

Course content: breakdown of the topics	Contact hours						Individual work: time and assignments		
	Lectures	Tutorials	Seminars	Practice	Laboratory work (LW)	Tutorial during LW	Contact hours	Individual work	Assignments
1. Course overview. Acquaintance with C# programming language. Applications build tools, .NET framework compatibility with different operating systems. C# Overview for programmers with OP knowledge. Code versioning systems.	2				2	8	4	10	Self-study of literature to deepen the knowledge. Preparation for laboratory works.
2. Type systems. Automatic and manual code review. Continuous integration. Applications deployment process. Improvement of software systems by using .NET technologies.	2				2		4	12	
3. Data types hierarchy. Classes, their structure and relationships. Generic types and methods. Conversions. Standard .NET interfaces. Creation of objects. Object lifecycle. Typical OOP mistakes and how to avoid them. Dependency Injection.	3				2		7	15	
4. Software system construction. Key goals and challenges. Business needs analysis. Software system modification and maintenance.	3				2		5	8	
5. C#-specific and OOP-specific properties. Delegates, anonymous types, lambda expressions. Events. Exceptions and their handling. Reflection, dynamic typing.	4				2		6	15	
6. Introduction to LINQ. Working with data. Data input and output, validation. Collections.	4				2		8	12	
7. Working with databases. Introduction to ORM. Overview of main ORM's. Entity Framework Core ORM implementation in .NET Core applications. LINQ use cases when working with databases. Serialization.	4				4		8	15	
8. Introduction to project management . Basics of Agile.	4				2		6	15	
9. Web services in .NET framework. REST, WCF, GraphQL and gRPC overview and examples.	4				3		8	15	
10. Introduction to multithreading. Async/Await. Real world examples.	4				2		8	12	
11. Ensuring the operation of the developed software. Unit and integration tests - principles, terminology. Tests in .NET environment.	4				4		8	15	
12. Functional and non-functional requirements for software systems. Use cases of interceptor and middleware in .NET Core framework.	2				2		4	8	
13. Introduction to creating and improving the user experience. Introduction to graphical interface development. Presentation of the developed software system.	3				1		4	10	
14. Overview of .NET technologies. Introduction to design patterns (MQ, CQRS etc.). Analysis of modern OO systems.	3				2		12	10	
15. Preparation for the exam and taking the final exam (written).		2					4	16	
<b>Total</b>	<b>48</b>	<b>2</b>			<b>32</b>	<b>8</b>	<b>82</b>	<b>188</b>	

Assessment strategy	Weight %	Deadline	Assessment criteria
Laboratory assignment No. 1	15	Week 8	<p>The collaborative laboratory work assigned to the students covers that the knowledge and skills that were developed in 1-7 topics. Student teams for collaborative work are recommended to be from 3 to 5 students, and for them not to rotate during the semester.</p> <p>All laboratory assignments must be placed in code repository and all the code must be reviewed by team mates.</p> <p>Additional points are added to the assessment if the work is presented before the deadline (no more than 10% of the final assessment and 5% for every preliminary week). Lateness leads to the decrease of the maximal assessment (1.5) by 20% of every delayed week.</p> <p>Each student in the team is evaluated separately, according to the student's responses during the review of the assignment, according to student's activeness during the previous laboratories, according to code reviews and created code scope. Partially finished laboratory work evaluated accordingly.</p>
Laboratory assignment No. 2	20	Week 12	<p>The collaborative laboratory work is continued, covering the knowledge and skills that were developed in 8-10 topics.</p> <p>All laboratory assignments must be placed in code repository and all the code must be reviewed by teammates.</p> <p>Additional points are added to the assessment if the work is presented before the deadline (no more than 10% of the final assessment and 5% for every preliminary week). Lateness leads to the decrease of the maximal assessment (2.0) by 20% of every delayed week.</p> <p>Each student in the team is evaluated separately, according to the student's responses during the review of the assignment, according to student's activeness during the previous laboratories, according to code reviews and created code scope. Partially finished laboratory work evaluated accordingly.</p>
Laboratory assignment No. 3	15	Week 15	<p>The individual laboratory work assigned to the students covers the knowledge and skills that were developed in 11-13 topics.</p> <p>Assignments require using databases.</p> <p>All laboratory assignments must be placed in code repository and all the code must be reviewed by teammates.</p> <p>Additional points are added to the assessment if the work is presented before the deadline (no more than 10% of the final assessment and 5% for every preliminary week). Lateness leads to the decrease of the maximal assessment (1.5) by 20% of every delayed week.</p> <p>Each student in the team is evaluated separately, according to the student's responses during the review of the assignment, according to student's activeness during the previous laboratories, according to code reviews and created code scope. Partially finished laboratory work evaluated accordingly.</p>
Mini quizzes during lectures	0-10	During the semester	During the lecture students might get asked random questions in an interactive way, which are not mandatory. Each answer can be rated from 0.01 to 0.05.
Exam in written form	50	Exam session	<p>Exam can be taken only when total amount of points collected during the semester is 3.0 or more.</p> <p>Maximum 5 points can be collected, which attribute to the 50% of the final score. The exam consists of 20 open, semi-open and close-ended questions and tasks each of them is assessed between 0.1 and 2 points (accordingly to the difficulties). Questions and tasks are formulated from topics set out in lectures. Exam is considered to be passed if at least 1.5 out of 5 points are collected.</p>

Requirements for subject evaluation by external method	
--	--

Evaluation is possible externally:	Yes
------------------------------------	-----

The student must have met the requirements for taking the exam. Previously earned points for work during the semester are credited. The student only takes the exam.

Author	Publis hing year	Title	Number or volume	Publisher or URL
<b>Required reading</b>				
Andrew Troelsen	2020	Pro C# 8 with .NET Core	9th ed.	Apress
Jon Skeet	2019	C# in Depth	4th ed.	Manning Publications
Andy Hunt	2019	The Pragmatic Programmer: your journey to mastery, 20th Anniversary Edition	2nd ed.	Addison-Wesley Professional
<b>Recommended reading</b>				
Tiberiu Covaci, Rod Stephens, Vincent Varallo, Gerry O'Brien	2013	MCS D Certification Toolkit (Exam 70-483)		
Dan Clark	2013	Beginning C# Object-Oriented Programming	2nd ed.	Apress
Jack Purdum	2012	Beginning Object-Oriented Programming with C#		Wiley / Wrox
Andrew Hunt, David Thomas	1999	The pragmatic programmer: from journeyman to master	1st ed.	The Pragmatic Bookshelf
Scott Allen	2015	C# Fundamentals with Visual Studio 2015	-	<a href="https://app.pluralsight.com/library/courses/c-sharp-fundamentals-with-visual-studio-2015/table-of-contents">https://app.pluralsight.com/library/courses/c-sharp-fundamentals-with-visual-studio-2015/table-of-contents</a>
James Shore	2007	The Art of Agile Development: Pragmatic Guide to Agile Software Development 1st Edition	-	O'Reilly Media, Inc.
Jake Knapp, John Zeratsky, Braden Kowitz	2016	Sprint: How to Solve Big Problems and Test New Ideas in Just Five Days	1st ed,	Simon & Schuster