COURSE SYLLABUS AND COURSE REQUIREMENTS
ACADEMIC YEAR 2025 SEMESTER 2

Course title	**Software Engineering**
Course Code	**IVB307ANMI**
Hours/Week: le/pr/lab	**2/0/0**
Credits	**2**
Degree Programme	***Computer Science Engineering BSc***
Study Mode	***Full-time***
Requirements	**Tests**
Teaching Period	**5**
Prerequisites	**Programming 3**
Department(s)	**Rendszer és Szoftvertechnológiai tanszék**
Course Director	***Dr. Iványi Péter***
Teaching Staff	***Varga Bálint***

COURSE DESCRIPTION

A short description of the course (max. 10 sentences).
Neptun: Instruction/Subjects/Subject Details/Basic data/Subject description

The course provides an overview of the fundamentals of Software Technology. It introduces students to the software development lifecycle, its phases, methodologies, and applied techniques. Students will also become familiar with the UML modeling language and the basics of system design

SYLLABUS

Neptun: Instruction/Subjects/Subject Details/Syllabus

1. GOALS AND OBJECTIVES

Goals, student learning outcome.
Neptun: Instruction/Subjects/Subject Details/Syllabus/Goal of Instruction

The aim of the course is to provide an overview of the fundamentals of Software Technology. Students will learn the stages of the software development lifecycle, become familiar with its phases, methodologies, and applied techniques. They will also acquire knowledge of the UML modeling language, various graphical models, and their applications. The course aims to teach the basics of system design and the application of modeling techniques.

2. COURSE CONTENT

Neptun: Instruction/Subjects/Subject Details/Syllabus/Subject content

	TOPICS
LECTURE	1. *Software Technology as a subject Principles of Software Development Tools of Software Technology and characteristics of a software development project* 2. *Software Development Lifecycle, models of software production, and phases of software development* 3. *Requirements and Requirements Specification, along with requirements gathering techniques*

4. *API Design and Service-Based Systems (including REST and GraphQL)*
5. *Data Model Design and Query Path Modeling*
6. *AI Tools in Software Development (such as code generation, refactoring, and testing support)*
7. *Software Architectures (comparing Monolithic vs. Microservices and exploring event-driven architecture)*
8. *Testing (covering verification, validation, the testing process, and test-driven development)*
9. *The Role, Types, and Creation of Documentation*
10. *Agile Methodologies (including SCRUM and Extreme Programming)*

DETAILED SYLLABUS AND COURSE SCHEDULE

ACADEMIC HOLIDAYS INCLUDED

LECTURE

week	**Topic**	**Compulsory reading; page number (from … to …)**	**Required tasks (assignments, tests, etc.)**	**Completion date, due date**
1.	*Software Technology as a subject Principles of Software Development Tools of Software Technology and characteristics of a software development project*	…	…	…
2	*Software Development Lifecycle, models of software production, and phases of software development*			
3.	*Requirements and Requirements Specification, along with requirements gathering techniques*			
4.	*API Design and Service-Based Systems (including REST and GraphQL)*			
5.	*Data Model Design and Query Path Modeling*			

6.	*Test 1*			
7.	*AI Tools in Software Development (such as code generation, refactoring, and testing support)*			
8.	*Software Architectures (comparing Monolithic vs. Microservices and exploring event-driven architecture)*			
9.	*Testing (covering verification, validation, the testing process, and test-driven development)*			
10.	*The Role, Types, and Creation of Documentation*			
11.	*Agile Methodologies (including SCRUM and Extreme Programming)*			
12.	Test 2			
13.	Retake test			

3. ASSESSMENT AND EVALUATION

(Neptun: Instruction/Subjects/Subject Details/Syllabus/Examination and Evaluation System)

ATTENDANCE

In accordance with the Code of Studies and Examinations of the University of Pécs, Article 45 (2) and Annex 9. (Article 3) a student may be refused a grade or qualification in the given full-time course if the number of class absences exceeds 30% of the contact hours stipulated in the course description.

Method for monitoring attendance *(e.g.: attendance sheet / online test/ register, etc.)*

Attendance of at least 70% of the classes is mandatory. The attendance rate does not affect the final grade, but absences exceeding 30% will result in failing the course.

Attendance is monitored based on an attendance sheet or short online test questions.

ASSESSMENT

Cells of the appropriate type of requirement is to be filled out (course-units resulting in mid-term grade or examination). Cells of the other type can be deleted.

***Course resulting in mid-term grade** (PTE TVSz 40§(3))*

***Mid-term assessments, performance evaluation and their ratio in the final grade** (The samples in the table to be deleted.)*

Type	Assessment	Ratio in the final grade
Test 1	*Percentage-based (earned points / maximum points * 100)*	*50%*
Test 2	*Percentage-based (earned points / maximum points * 100)*	*50%*

Opportunity and procedure for re-takes (PTE TVSz 47§(4))

The specific regulations for improving grades and resitting tests must be read and applied according to the general Code of Studies and Examinations. E.g.: all tests and assessment tasks can be repeated/improved at least once every semester, and the tests and home assignments can be repeated/improved at least once in the first two weeks of the examination period.

If the weighted average of the exams is below 40%, a make-up exam must be taken once, covering the entire semester's material.

Make-up exam: Week 13

Grade calculation as a percentage

based on the aggregate performance according to the following table

Course grade	Performance in %
excellent (5)	85 % …
good (4)	70 % ... 85 %
satisfactory (3)	55 % ... 70 %
pass (2)	40 % ... 55 %
fail (1)	below 40 %

The lower limit given at each grade belongs to that grade.

4. SPECIFIED LITERATURE

In order of relevance. (In Neptun ES: Instruction/Subject/Subject details/Syllabus/Literature)

COMPULSORY READING AND AVAILABILITY

[1.] Ian Sommerville, Software Engineering, 10th Edition, Pearson, 2015 (https://dn790001.ca.archive.org/0/items/bme-vik-konyvek/Software%20Engineering%20-%20Ian%20Sommerville.pdf)
[2.] The lecture materials uploaded by the instructor to Neptun.

RECOMMENDED LITERATURE AND AVAILABILITY

[3.] R.S. Pressmann: Software Engineering, a Practition's approach, 7th Edition, McGraw-Hill Higher education, 2010.
[4.] Jeffry L. Whitten, Lonnie D. Bentley: Systems Analysis and Design Methods, 7th Edition, 2007, McGraw-Hill