

## COURSE SYLLABUS AND COURSE REQUIREMENTS

### ACADEMIC YEAR 2024/2025 SEMESTER I

<i>Course title</i>	<i>Programming 2</i>
<i>Course Code</i>	IVB054ANMI
<i>Hours/Week: le/pr/lab</i>	2 / 0 / 2
<i>Credits</i>	5
<i>Degree Programme</i>	<i>Computer Science Engineering BSc</i>
<i>Study Mode</i>	<i>Full-time</i>
<i>Requirements</i>	Term mark
<i>Teaching Period</i>	2024-25/1
<i>Prerequisites</i>	Programming 1
<i>Department(s)</i>	Systems and Software Technologies
<i>Course Director</i>	<i>Dr. Storz, Tamas</i>
<i>Teaching Staff</i>	<i>Dr. Storz, Tamas</i>

## COURSE DESCRIPTION

*A short description of the course (max. 10 sentences).*

*Neptun: Instruction/Subjects/Subject Details/Basic data/Subject description*

Students would see generalization and transition of their previously acquired programming knowledge.

Participants of the course could meet most common programming paradigm of nowadays. After understanding its principals, they can learn its proper application with appropriate tools to create and maintain sustainable source code.

The iterative application of OOP principles is done in the most popular environment: Java.

During the semester, while practicing programming skills, students learn to professionally evaluate own work and see teamwork tools.

## 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 purpose of this course is to make students understand and able to utilize fundamental concepts and tools of object-oriented programming in Java environment for creation and maintenance of properly working application with sustainable source codes.

### 2. COURSE CONTENT

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

## TOPICS

<b>LECTURE</b>	<ol style="list-style-type: none"><li>1. <i>Java platform history and structure</i></li><li>2. <i>Object Oriented Programming tools</i></li><li>3. <i>SOLID principals and their application</i></li><li>4. <i>Extras of Methods</i></li></ol>
<b>PRACTICE</b>	<ol style="list-style-type: none"><li>1. <i>IDE, project, debugging, git</i></li><li>2. <i>Procedural programming refresh</i></li><li>3. <i>Object Oriented Programming</i></li><li>4. <i>Extras of Methods</i></li></ol>

## 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.	Functional and procedural programming	#1 / 1		
2.	Java history, platform structure, comparison of basic elements of C and Java language	#1 / 2		
3.	Object oriented programming – encapsulation, often used types	#1 / 3		
4.	Class ( <i>static</i> ) components, arrays, often used classes, handling the console	#1 / 4	Homework	w#8 (Sunday 23:59)
5.	Exception handling, text file management, serialization	#1 / 5		
6.	Methods in detail, Inheritance, Polymorphism, Class cast	#1 / 6		
7.	Abstract classes	#1 / 7		
8.	Interfaces, cohesion and decoupling	#1 / 8		
9.	Autumn break		Homework retake	w#10 (Sunday 23:59)
10.	Type parameter, generic programming	#1 / 9		
11.	Application design and implementation			
12.	Method reference, anonymous, lambda	#1 / 10		
13.	Questions and answers			
14.	Questions and answers			

### PRACTICE, LABORATORY PRACTICE

week	Topic	Compulsory reading; page number (from ... to ...)	Required tasks (assignments, tests, etc.)	Completion date, due date
1.	Java, IntelliJ IDEA, git, functional programming			
2.	Programming theses			
3.	Source code refactoring, class/object creation			
4.	Static components, arrays, console			
5.	Exceptions, text files			
6.	Method override, visibility, inheritance, polymorphism			
7.	Abstract methods, classes,			
8.	Interface declaration, implementation			
9.	Autumn break			
10.	Type parameter, generic programming			
11.	Application design and implementation practice		participation	
12.	Method references, anonymous, lambda			
13.	Theoretical and practical exam		Complex exam	
14.	Theoretical and practical exam retake		Complex exam	

### 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 sheet, on-line test

#### 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.)

Assessment points are multiplied by the homework/exam result percentage accordingly.

None of the assignments is mandatory, any combination can be selected. Assignment results are sum up, except when a practice exam is taken, its result REPLACES the homework result. In this case, homework is only practice for the exam.

Type	Assessment	Ratio in the final grade
Homework OR Practice exam	100%	60% or 80%
Theory test	100%	20 %
<b>Total</b>		<b>80% or 100 %</b>

**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.

Homework: resubmission on week #10

Tests: retake on 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.] Course slides

[2.] Course practice examples

#### RECOMMENDED LITERATURE AND AVAILABILITY

[3.] Bruce Eckel: Thinking in Java, (2006) Pearson, ISBN 978-0131872486