



PROGRAMMING 1. OBJECT ORIENTED PROGRAMMING AND DESIGN PATTERNS

Syllabus

Requisites of the Course

Cycle of Higher Education	<i>First cycle of higher education (Bachelor's degree)</i>
Field of Study	<i>12 Information Technologies</i>
Speciality	<i>121 Software engineering</i>
Education Program	<i>Software Engineering of Multimedia and Information Retrieval Systems</i>
Type of Course	<i>Normative</i>
Mode of Studies	<i>full-time</i>
Year of studies, semester	<i>2 year (3 semester)</i>
ECTS workload	<i>4 credits (ECTS). Time allocation: 36 hours for lectures, 18 hours for programming assignments, 54 hours for self-study.</i>
Testing and assessment	<i>Exam</i>
Course Schedule	<i>According to rozklad.kpi.ua</i>
Language of Instruction	<i>English</i>
Course Instructors	<i>Senior lecturer, Olga Sulema, PhD olga.sulema@pzks.fpm.kpi.ua</i>
Access to the course	<i>Google classroom at https://classroom.google.com/u/0/c/MzKxODYxOTUzMzAy</i>

Outline of the Course

1. Course description, goals, objectives, and learning outcomes

The study of the Programming 1. Object Oriented Programming and Design Patterns course allows students acquire competencies necessary for solving practical problems related to the designing and development of object-oriented software.

The purpose of studying the Programming 1. Object Oriented Programming and Design Patterns course is to build capacity to choose and implement the principles of OOP as well as design pattern for efficient programming code development.

The subject of the Programming 1. Object Oriented Programming and Design Patterns course is the process of software development.

2. Prerequisites and post-requisites of the course (the place of the course in the scheme of studies in accordance with curriculum)

The Programming 1. Object Oriented Programming and Design Patterns course is a normative discipline and students do not need any specific initial knowledge for its study.

Theoretical knowledge and practical skills acquired in the Programming 1. Object Oriented Programming and Design Patterns course provide the necessary background for studying other disciplines in Bachelor and Master programs of 121 Software Engineering specialty.

3. Content of the course

Topic 1. Introduction to C# and .NET

Topic 2. Fundamental principles of OOP

Topic 3. Software development with OOP

Topic 4. Structural design patterns

Topic 5. Creational design patterns

Topic 6. Behavioral design patterns

Topic 7. Software development techniques

4. Coursebooks and teaching resources

Main literature:

1. Taher R. *Hands-On Object-Oriented Programming with C#, 2019.*
2. Perkins B., Hammer J.V., Reid J.D. *Beginning C# 7 Programming with Visual Studio, 2017.*
3. Olsson M. *C# 7 Quick Syntax Reference: A Pocket Guide to the Language, APIs, and Library, 2018.*
4. Shvets A. *Dive Into Design Patterns, 2020.*
5. Bender J., McWherter J. *Professional Test-Driven Development with C#: Developing, 2011.*
6. Bishop J. *C# 3.0 Design Patterns, 2008.*

Educational content

5. Methodology

No	Type of a class	Materials for self-studying
<i>Topic 1. Introduction to C# and .NET</i>		
1.	<i>Lecture 1. Introduction to C#</i>	<i>6, №1</i>
2.	<i>Lecture 2. .NET Platform</i>	<i>6, №2</i>
3.	<i>Lecture 3. Basics of C# programming. Part 1</i>	<i>6, №3</i>
4.	<i>Lecture 4. Basics of C# programming. Part 2</i>	<i>6, №4</i>
5.	<i>Programming assignment 1. Console application in C#</i>	<i>6, №5</i>
<i>Topic 2. Fundamental principles of OOP</i>		
6.	<i>Lecture 5. Classes and objects</i>	<i>6, №6</i>
7.	<i>Lecture 6. The four principles of OOP</i>	<i>6, №7</i>
8.	<i>Lecture 7. Inheritance and encapsulation in C#</i>	<i>6, №8</i>
9.	<i>Lecture 8. Abstraction in C#</i>	<i>6, №9</i>
10.	<i>Lecture 9. Polymorphism in C#</i>	<i>6, №10</i>
11.	<i>Programming assignment 2. Implementation of inheritance and encapsulation in a C# application</i>	<i>6, №11</i>
12.	<i>Programming assignment 3. Implementation of abstraction and polymorphism in a C# application</i>	<i>6, №12</i>
<i>Topic 3. Software development with OOP</i>		
13.	<i>Lecture 10. Types of object collaboration</i>	<i>6, №13</i>
14.	<i>Lecture 11. Exception handling</i>	<i>6, №14</i>

15.	<i>Lecture 12. Generics</i>	6, №15
16.	<i>Programming assignment 4. Development of a C# application</i>	6, №16
<i>Topic 4. Design patterns</i>		
17.	<i>Lecture 13. Introduction to design patterns</i>	6, №17
18.	<i>Lecture 14. Structural design patterns</i>	6, №18
19.	<i>Lecture 15. Creational design patterns</i>	6, №19
20.	<i>Lecture 16. Behavioral design patterns</i>	6, №20
21.	<i>Lecture 17. The MVC pattern</i>	6, №21
22.	<i>Programming assignment 5. Implementation of design patterns in a C# application</i>	6, №22
23.	<i>Midterm test</i>	6, №23

6. Self-study

No	Topic for self-studying	Hours	Literature
1.	<i>Preparation to a lecture 1</i>	1	1, 5
2.	<i>Preparation to a lecture 2</i>	1	1, 3, 5
3.	<i>Preparation to a lecture 3</i>	1	1, 2
4.	<i>Preparation to a lecture 4</i>	1	1, 2, 5
5.	<i>Preparation to a programming assignment 1</i>	1,5	1, 2, 3, 5
6.	<i>Preparation to a lecture 5</i>	1	1, 3
7.	<i>Preparation to a lecture 6</i>	1	1, 3
8.	<i>Preparation to a lecture 7</i>	1	1, 2, 3
9.	<i>Preparation to a lecture 8</i>	1	2
10.	<i>Preparation to a lecture 9</i>	1	2, 3
11.	<i>Preparation to a programming assignment 2</i>	1,5	1, 2, 3
12.	<i>Preparation to a programming assignment 3</i>	1,5	1, 2, 3
13.	<i>Preparation to a lecture 10</i>	1	1
14.	<i>Preparation to a lecture 11</i>	1	1, 3
15.	<i>Preparation to a lecture 12</i>	1	1, 3
16.	<i>Preparation to a programming assignment 4</i>	1,5	
17.	<i>Preparation to a lecture 13</i>	1	4, 6
18.	<i>Preparation to a lecture 14</i>	1	4, 6
19.	<i>Preparation to a lecture 15</i>	1	4, 6

20.	<i>Preparation to a lecture 16</i>	1	4, 6
21.	<i>Preparation to a lecture 17</i>	1	4, 6
22.	<i>Preparation to a programming assignment 5</i>	1,5	1, 4, 5
23.	<i>Preparation to a midterm test</i>	4	1, 5
24.	<i>Preparation to an exam</i>	6	1, 5
25.	<i>Basics of C#</i>	4	1, 2, 3
26.	<i>Principles of OOP</i>	4	1
27.	<i>Structural design patterns</i>	4	4, 6
28.	<i>Creational design patterns</i>	4	4, 6
29.	<i>Behavioral design patterns</i>	4	4, 6

Policy and Assessment

7. Course policy

- *Attending lectures is mandatory.*
- *Attending seminars is mandatory.*
- *Rules of conduct: activity, taking part in discussions, respect to teacher and groupmates, muting cellphones.*
- *Compliance with the policy of academic integrity.*
- *Rules of fulfilling seminar tasks: assignment has to be completed according to the student's team number or the student's variant defined with a generator of pseudo-random numbers (hereafter – randomizer).*
- *Rules of assigning bonus points and penalty points:*

Bonus points are being assigned for:

- *taking part in discussion during lecture classes;*
- *answering other students' questions on lecture material;*
- *creative approach in team tasks.*

Max bonus points during the semester: 5 points.

Penalty points might be assigned because of:

- *plagiarism: -5 points for every attempt;*
- *turning programming assignments in untimely: -0,5 points for every week after the deadline (max penalty points for one programming assignment: -5 points).*

8. Monitoring and grading policy

In the first lecture, the students are being acquainted with the grading policy which is based on Regulations on the System of Learning Outcomes Assessment (https://document.kpi.ua/files/2020_1-273.pdf).

The student's rating in the course consists of points that they receive for programming assignment (R_1), a midterm test (R_2) and an exam (R_3).

$$R_5 = R_1 + R_2 + R_3 = 35 + 25 + 40 = 100 \text{ points}$$

According to the university regulations on the monitoring of students' academic progress (https://kpi.ua/document_control), there are two assessment weeks, usually during 7th/8th and 14th/15th week of the semester, when students take the Progress and Module tests respectively, to check their progress against the criteria of the course assessment policy.

The table of compliance between overall points and the final grade:

Points	Grade
95-100	Excellent
85-94	Very good
75-84	Good
64-74	Satisfactory
60-64	Fair
Less than 60	Unsatisfactory
Course requirements are not met	Not Graded

9. Additional information about the course

The list of questions for an exam are adduced in Appendix 1.

Syllabus of the course

Is designed by PhD, senior lecturer, Olga Sulema

Adopted by Computer Systems Software Department (protocol № 12, 26 April 2023)

Approved by the Faculty Board of Methodology (protocol № 10, 26 May 2023)