OBJECT-ORIENTED PROGRAMMING. INTRODUCTORY PRACTICE

COURSE SYLLABUS

Code and name of specialty	121 Software Engineering	**Institute / faculty**	Faculty of Computer Science and Software Engineering
Program name	"Software Engineering"	**Department**	Software Engineering and Management Information Technologies
Type of program	Educational and Professional	**Language of instruction**	Ukrainian, English

LECTURER

Full name, e-mail	**Lev Ivanov, *Lev.Ivanov@khpi.edu.ua***

Senior Lecturer at the Department of Software Engineering and Management Information Technologies. Work experience – since 1981. Author (co-author) of more than 30 research papers and textbooks (Google Scholar: https://scholar.google.com/citations?user=ADPHLAsAAAAJ).

Leading lecturer of the courses: "Programming Basics (Part 1, Part 2) (English and Ukrainian)", "Object-oriented programming. Introductory practice" (English and Ukrainian), "Fundamentals of Java programming" (English and Ukrainian).

GENERAL DESCRIPTION OF THE COURSE

Summary	The course "Object-oriented programming. Introductory practice" is a course in the cycle of professional compulsory training of the specialty 121 "Software Engineering". It is taught in the third semester in the amount of 120 hours (4 ECTS credits), in particular: lectures – 32 hours, laboratory classes – 32 hours, independent work – 56 hours. The course includes two content modules and one course work. The study of the discipline ends with the exam.
Course objectives	Assimilation of the necessary knowledge of mastering the theoretical bases of programming languages C++ and Java and the acquisition of practical skills of their use during the development of programs based on the principles of structural, procedural-oriented and object-oriented programming.
Types of classes and control	Lectures, laboratory classes. Continuous assessment – laboratory works, intermediate modular assessment, course work. Final assessment – exam.
Term	3

Student workload (credits) / Type of course	4 / Mandatory	**Lectures (hours)**	32	**Workshops (hours)**	32	**Self-study (hours)**	56

Program competences	GC02. Ability to apply knowledge in practical situations. GC05. Ability to learn and master modern knowledge. GC06. Ability to search, process and analyze information from various sources. GC 07. Ability to work in a team. PC14. Ability to participate in software design, including modelling (formal description) of its structure, behavior and functioning processes. PC15. Ability to develop architectures, modules and components of software systems. PC22. Ability to accumulate process and systematize professional knowledge on the creation and maintenance of software and recognition of the importance of lifelong learning. PC25. Ability to reasonably select and master software development and maintenance tools.

Learning outcomes	Teaching and learning methods	Forms of assessment (continuous assessment CAS, final assessment FAS)
PO01. Analyze, purposefully search for and select the necessary information and reference resources and knowledge to solve professional problems, taking into account modern advances in science and technology.	Interactive lectures with presentations, course work, discussions, laboratory classes, teamwork, problem learning	Individual tasks for laboratory work (CAS), assessment of knowledge in laboratory classes (CAS), express survey (CAS),course work (CAS), final/semester control in the form of a semester exam, in accordance with the schedule of the educational process (FAS)
PO04. Know and apply professional standards and other regulatory documents in the field of software engineering.	Interactive lectures with presentations, course work, discussions, laboratory classes, teamwork, problem learning	Individual tasks for laboratory work (CAS), assessment of knowledge in laboratory classes (CAS), express survey (CAS),course work (CAS), final/semester control in the form of a semester exam, in accordance with the schedule of the educational process (FAS)
PO05. Know and apply relevant mathematical concepts, methods of domain, system and object-oriented analysis and mathematical modeling for software development.	Interactive lectures with presentations, course work, discussions, laboratory classes, teamwork, problem learning	Individual tasks for laboratory work (CAS), assessment of knowledge in laboratory classes (CAS), express survey (CAS),course work (CAS), final/semester control in the form of a semester exam, in accordance with the schedule of the educational process (FAS)
PO07. Know and apply in practice the fundamental concepts, paradigms and basic principles of operation of language, tools and computing software engineering.	Interactive lectures with presentations, course work, discussions, laboratory classes, teamwork, problem learning	Individual tasks for laboratory work (CAS), assessment of knowledge in laboratory classes (CAS), express survey (CAS),course work (CAS), final/semester control in the form of a semester exam, in accordance with the schedule of the educational process (FAS)
PO08. Be able to develop a human-machine interface	Interactive lectures with presentations, course work, discussions, laboratory classes, teamwork, problem learning	Individual tasks for laboratory work (CAS), assessment of knowledge in laboratory classes (CAS), express survey (CAS),course work (CAS), final/semester control in the form of a semester exam, in accordance with the schedule of the educational process (FAS)
PO13. Know and apply methods of algorithm development, software design and data and knowledge structures.	Interactive lectures with presentations, course work, discussions, laboratory classes, teamwork, problem learning	Individual tasks for laboratory work (CAS), assessment of knowledge in laboratory classes (CAS), express survey (CAS),course work (CAS), final/semester control in the form of a semester exam, in accordance with the schedule of the educational process (FAS)
PO14. Put into practice the tools of domain analysis, design, testing, visualization, measurement and documentation of software.	Interactive lectures with presentations, course work, discussions, laboratory classes, teamwork, problem learning	Individual tasks for laboratory work (CAS), assessment of knowledge in laboratory classes (CAS), express survey (CAS),course work (CAS), final/semester control in the form of a semester exam, in accordance with the schedule of the educational process (FAS)
PO16. Have the skills of team development, approval, design and release of all types of software documentation.	Interactive lectures with presentations, course work, discussions, laboratory classes, teamwork, problem learning	Individual tasks for laboratory work (CAS), assessment of knowledge in laboratory classes (CAS), express survey (CAS),course work (CAS), final/semester control in the form of a semester exam, in accordance with the schedule of the educational process (FAS)
PO17. Be able to apply methods of component software development.	Interactive lectures with presentations, course work, discussions, laboratory classes, teamwork, problem learning	Individual tasks for laboratory work (CAS), assessment of knowledge in laboratory classes (CAS), express survey (CAS),course work (CAS), final/semester control in the form of a semester exam (FAS)

PO23. Be able to document and present the results of software development.	Interactive lectures with presentations, course work, discussions, laboratory classes, teamwork, problem learning	Individual tasks for laboratory work (CAS), assessment of knowledge in laboratory classes (CAS), express survey (CAS),course work (CAS), final/semester control in the form of a semester exam, in accordance with the schedule of the educational process (FAS)

ASSESSMENT AND GRADING

	score (points) for all types of learning activities	ECTS grading scale	The national grading scale
Ranges of points correspon-ding to grades	90-100	A	excellent
	82-89	B	good
	74-81	C	
	64-73	D	satisfactory
	60-63	E	
	35-59	FX	Unsatisfactory (with the exam retake option)
	0-34	F	Unsatisfactory (with mandatory repetition of the course)

Allocation of grade points

100% Final assessment as a result of Final exam (20%) and Continuous assessment (80%).
20% Final exam: semester exam, according to the schedule of the educational process
80% Continuous assessment:
• 50% of assessment of tasks in laboratory works;
• 10% intermediate control (colloquium)
• 20% Course work

Course policy

Students must attend all classes according to the study schedule and adhere to the norms of academic ethics. To study the course, students need to have their personal computer and (or) use computers of the computer center at the department. Students must work with compulsory and recommended reading, including Internet resources. Students must complete and submit all laboratory works during the semester in which the course is taught, before the examination session. The final assessment is not carried out without the personal presence of students.

COURSE STRUCTURE AND CONTENT

Topic 1	Basic syntax of C#. Encapsulation and inheritance	**Laboratory work 1**	Creation and use classes in C # Innovation Campus: П325 – ООР – Sprint05	**Self-study**	Work with properties and indexers.
Topic 2	Polymorphism and interfaces	**Laboratory work 2**	Creation class hierarchies Innovation Campus: П326 – ООР – Sprint06		Class hierarchy and object hierarchy
Topic 3	Work with exceptions, generalizations, and collections	**Laboratory work 3**	Using container classes Innovation Campus П327 – ООР – Sprint07		Disadvantages and advantages of using exceptions
Topic 4	Event-driven programming. Functional and declarative programming	**Laboratory work 4**	GUI Application Development Innovation Campus: П328 – ООР – Sprint08		Delegates and events
Topic 5	Using design patterns	**Laboratory work 5**	Applying design patterns in C# applications		Factory method and abstract factory

			Innovation Campus: П329 – ООР – Sprint09		

RECOMMENDED READING

Compulsory

1. Troelsen, A., Japikse, P. (2021). Pro C# 9 with .NET 5: Foundational Principles and Practices in Programming. (10th ed.). Apress.
2. Коноваленко, І. В., Марущак П. О., Савків В. Б. (2017). Програмування мовою C# 7.0 : навчальний посібник. Тернопіль :Тернопільський національний технічний університет імені Івана Пулюя.
3. Price, M. J. (2020). C# 9 and .NET 5-Modern Cross-Platform Development: Build intelligent apps, websites, and services with Blazor, ASP.NET Core, and Entity Framework Core using Visual Studio Code.(2020). (5th ed.). Packt Publishing.
4. Albahari, J. (2021). C# 9.0 in a Nutshell: The Definitive Reference. (1st ed.). O'Reilly Media.
5. Nagel, C. (2018). Professional C# 7 and .NET Core 2.0. (7th ed.) Wrox.
6. Cardoso, A. F. M. (2021). Implementing Design Patterns in C# and .NET 5: Build Scalable, Fast, and Reliable .NET Applications Using the Most Common Design Patterns. BPB Publications.
7. Booch, G., Rumbaugh, J., Jacobson, I. (2005). The Unified Modeling Language User Guide (Object Technology Series). (2nd ed.). Addison-Wesley Professional.
8. Gamma, E., Helm, R., Johnson, R., Vlissides, J. (1994). Design Patterns: Elements of Reusable Object-Oriented Software. (1st ed.). Addison-Wesley Professional.
9. Weisfeld, M. (2019). Object-Oriented Thought Process, The Developer's Library. (5th ed.).

Recommended

1. Farrell, J. (2017). Microsoft Visual C#: An Introduction to Object-Oriented Programming.(7 ed.). Joyce Cengage Learning.
2. Дудзяний, І. М. (2007). Об'єктно-орієнтоване моделювання програмних систем: Навч. посібник. Львів: Видавничий Центр ЛНУ імені Івана Франка.
3. Голуб, Б. М. (2006). C#. Концепція та синтаксис. Навч. посібник. Львів: Видавничий центр ЛНУ імені Івана Франка.

INTERNET RESOURCES

1. C# Language Specification. Retrieved from http://msdn.microsoft.com/en-us/library/ ms228593.aspx (англ.)
2. C# Language Specification. Version 4.0. Retrieved from http://www.ecma-international.org/publications/files/ECMA-ST/Ecma-334.pdf
3. Philips Healthcare - C# Coding Standard. Version 2.0 Retrieved from http://www.tiobe.com/content/paperinfo/gemrcsharpcs.pdf
4. C# Coding Standards for .NET. Retrieved from http://se.inf.ethz.ch/old/ teaching/ss2007/251-0290-00/project/CSharpCodingStandards.pdf
5. Visual C#. Retrieved from http://msdn.microsoft.com/en-us/library/kx37x362
6. Visual C# Guided Tour. Retrieved from http://msdn.microsoft.com/en-us/library/ bb383962(v=vs.90).aspx
7. C# / CSharp Tutorial. Retrieved from http://www.java2s.com/Tutorial/CSharp/ CatalogC Sharp.htm
8. Learning C# by Example. Retrieved from http://www.fincher.org/tips/Languages /csharp.shtml
9. Огляд платформи .Net. Retrieved from http://www.articledb.org.ua/stattya/oglyad-platformy-net.htm
10. Мова програмування C#. Retrieved from http://www.znannya.org/?view=csharp
11. Підручник C# з прикладами і завданнями. Retrieved from http://emerecu.ukma. kiev.ua/offline_courses/course86/summary.htm

Academic integrity

Graduate students are expected to adhere to the Code of Ethics of Academic Relations and Integrity" of NTU "KhPI".

The content of this syllabus is consistent with the course program.