National Technical University of Ukraine
"Igor Sikorsky Kyiv Polytechnic Institute"

Computer Systems Software Department

Design of modern high-performance computing systems

Syllabus

Course Requisites	
Cycle of Higher Education	*Second (master's)*
Field of Study	*12 Information Technologies*
Speciality	*121 Software engineering*
Educational program	*Software Engineering of Multimedia and Information Retrieval Systems*
Discipline status	*Elective*
Study form	*Full-time*
Year of study, semester	*1 year 2 semester*
ECTS workload	*5 credits (ECTS). Time allotment - 150 hours, including 54 hours of classroom work, and 96 hours of self-study*
Testing and assessment	*exam*
Course Schedule	*Classes by the timetable http://rozklad.kpi.ua/*
Language	*English*
Course Instructors	*PhD., Associate Professor, Vasyl Yurchyshyn, vasylPZKS@gmail.com*
Access to the course	Google classroom, http://fpm.kpi.ua/archive

Outline of the Course

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

Studying the discipline "Design of modern high-performance computing systems" allows students to develop the competencies necessary for solving practical problems related to the development and use of modern high-performance computing systems in the everyday professional activities of a programmer.

The purpose *of studying the discipline " Design of modern high-performance computing systems" is to form in students the ability to independently develop software for the design of modern high-performance computing systems and their use, as well as to use third-party software to create modern high-performance computing systems.*

The subject *of the discipline " Design of modern high-performance computing systems" is modern methods and technologies of designing high-performance computing systems with the wide application of cloud, grid and supercomputer technologies of software direction.*

The study of the discipline " Design of modern high-performance computing systems" contributes to the formation of students of professional competences (FC) necessary for solving practical tasks of professional activity related to the development, improvement and operation of information systems:

Competencies, the formation of which is facilitated by this discipline:

FC 01	*Ability to analyze subject areas, form, classify software requirements.*

FC 02	*Ability to develop and implement scientific and/or applied projects in the field of software engineering*
FC 03	*The ability to design software architecture, to model the functioning processes of individual subsystems and modules*
FC 05	*Ability to develop, analyze and apply specifications, standards, rules and recommendations in the field of software engineering.*
FC 07	*Ability to think critically about problems in the field of information technology and at the boundaries of fields of knowledge, integrate relevant knowledge and solve complex problems in broad or multidisciplinary contexts.*
FC 08	*Ability to develop and coordinate processes, stages and iterations of the software life cycle based on the application of modern software development models, methods and technologies*
FC 13	*Ability to implement and maintain information systems.*
FC 16	*Ability to apply software engineering methodologies in practice*

The formation of the specified competencies contributes to the formation of the following program learning outcomes (PLO) in students according to the educational program:

PLO 02	*Evaluate and choose effective methods and models of software development, implementation, support and management of relevant processes at all stages of the life cycle*
PLO 03	*Build and research models of information processes in the applied field.*
PLO 04	*Identify information needs and classify data for software design*
PLO 06	*Develop and evaluate software design strategies; substantiate, analyze and evaluate options for project solutions from the point of view of the quality of the final software product, resource limitations and other factors.*
PLO 07	*Analyze, evaluate and apply modern software and hardware platforms at the system level to solve complex software engineering problems*
PLO 08	*Розроблятиі модифікувати архітектуру програмного забезпечення для реалізації вимог замовника.*
PLO 10	*Develop and modify software architecture to meet customer requirements.*
PLO 11	*Know the basic concepts and methodologies of modeling information processes*
PLO 13	*Configure software, manage its changes and development of software documentation at all stages of the life cycle.*
PLO 15	*Carry out software reengineering in accordance with customer requirements*
PLO 16	*Plan, organize and carry out software testing, verification and validation.*
PLO 17	*Collect, analyze, evaluate information necessary for solving scientific and applied problems, using scientific and technical literature, databases and other sources.*
PLO 18	*To know the theoretical principles underlying the research methods of information systems and software, the methodology of conducting research and computational experiments.*
PLO 19	*Know design technologies and methods of ensuring high performance of software systems*
PLO 29	*Conduct controlled calculations in GRID and cloud systems, provide protection for GRID services, develop data processing software in GRID and cloud services*
PLO 30	*To know the methods and instrumental means of modeling and designing information systems*
PLO 47	*Model and design web-oriented and enterprise software systems with optimized parameters*

2. Discipline prerequisites and postrequisites (place in the structural and logical education scheme according to the relevant educational program)

The successful study of the discipline "Design of modern high-performance computing systems" is preceded by the study of the disciplines "Programming", "Object-oriented programming", "Parallel and distributed computing" of the curriculum of bachelor's training in the specialty 121 Software engineering.

The theoretical knowledge and practical skills acquired during the mastering of the discipline "Design of modern high-performance computing systems" ensure the successful completion of course projects and master's theses in the specialty 121 Software engineering.

3. Content of the course

The discipline "Design of modern high-performance computing systems" involves studying the following topics:

Topic 1 Technologies, methods and means of designing high-performance computer systems.

Topic 2 Design standards and design documentation.

Topic 3. Design stages and stages.

Topic 4. Technologies for creating high-performance computing systems.

Topic 5. Instrumental means of designing high-performance systems.

Topic 6. CASE-technology

Topic 7. Grid and cloud environments for designing high-performance computing systems.

Topic 8 SOLIDWORKS design system.

Modular control work

Exam

4. Coursebooks and teaching resources

Basic references:

1. Frederick P. Brooks, Jr. The Mythical Man-Month. Essays on Software Engineering. Anniversary Edition, ADDISON WESLEY. — *2021. — 304 pp.* — ISBN 0-201-83595-9

2. Frederick P. Brooks, Jr. THE DESIGN OF DESIGN. Essays from a Computer Scientist, ADDISON WESLEY. — *2020. — 445 pp.* — ISBN 978-0-201-36298-5

3. Craig Larman. APPLYING UML AND PATTERNS. An Introduction to Object-Oriented Analysis and Design theUnified Process, Prentice Hall PTR, 2002. – 624 pp. — ISBN 0-213-092569-1

4. А. І. Дичка, В. Я. Юрчишин. Проєктування сучасних високопродуктивних обчислювальних систем: комп'ютерний практикум – Київ : КПІ ім. Ігоря Сікорського, 2022. – 87 с.

5. В.Я.Юрчишин. Хмарні та Грід-технології: конспект лекцій: навч. посіб. для студ. спеціальності 121 «Інженерія програмного забезпечення». – Київ : КПІ ім. Ігоря Сікорського, 2019. – 264 с.. *Матеріали знаходяться у вільному доступі в Інтернеті.*

6. Ambler. S. W. Agile Database Techniques: Effective Strategies for the Agile

Software Develo per.John Wiley & Sons, 2023. – 416 pp..

7. В. Я. Юрчишин. Проєктування сучасних високопродуктивних обчислювальних систем: Лекції – Київ : КПІ ім. Ігоря Сікорського, 2022. – 279 с.

8.Gomaa, Hassan. Software modeling and design : UML, use cases, patterns, and software architectures, Cambridge University Press, Langara College, 2019, p.578.
9. Foster, C. Kesselman, S. Tuecke, J.M. Nick. The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration. - Morgan Kaufmann Publishers, 2002.
10. Cloud Computing and Software Services. Theory and Techniques. / S. A. Ahson, M.Ilyas - ed-s.- CRC Press Taylor & Francis Group. – 2011, p.442.
11.Basis modeling concepts: Intro to 3D, AutoCAD to SolsdWorks, Parametric Modeling. Basis – http://www.solidprofessor.com/product-matrix.aps.

Educational content

5. Methods of mastering the discipline (educational component)

№	Training session type	Lesson description
Topic 1 Technologies, methods and means of designing high-performance computing systems.		
1	Lecture 1 Introduction to practical engineering design.	Fields of application. Basic concepts and definitions. Tasks Classification. Functions and requirements, means of designing high-performance systems. Task for independent work: item 6, number 1.
2	Lecture 2. Types of project work. Methods, technologies and means of designing high-performance information systems	IC design methodology. Concept of IS design method and technology. Classification of IS design methods. Design automation. Automated design systems. Task for independent work: item 6, number 2.
3	Laboratory work 1. Exploratory and instrumental NDR. Purposes, sieves, stages.	Task: To learn to formulate the goals and objectives of the NDR, to determine the main stages of their implementation and the typical composition of reporting documentation and the development of requirements and basic initial data for research and design development. Task for independent work: item 6, number 3.
Topic 2 Design standards and design documentation.		
4	Lecture 3. Classification and types of standards for designing information systems	Normative and methodical provision of information systems design. Types, composition and content of project and program documentation. Standards, guiding documents, methods, provisions, instructions, templates, regulatory documents. Task for independent work: item 6, number 4.
5	Laboratory work 2. Terms of reference of the project.	Task: To learn how to draw up a technical task in the established order, to determine the goals of creating a project, requirements and basic initial data necessary for development. Task for independent work: item 6, number 5.
Topic 3. Stages and stages of design.		
6	Lecture 4. Stages and stages of designing information systems	Procedure for system development, implementation and maintenance. The main stages and stages of designing information systems. Tasks on SRS: item 6, number 6.

7	*Laboratory work 3. Determination of the composition of the documentation of the draft project, technical project and the stage of development of the experimental sample.*	*Task: Create a list of documentation of the sketch, technical and working project based on current state regulatory documentation.* *Task for independent work: item 6, number 7.*
8	*Lecture 5 The process of creating an information system*	*Formulation of requirements, concepts, testing, commissioning, support of the information system.* *Tasks on SRS: item 6, number 8.*
9	*Laboratory work 4 Determination of the stages of IS creation*	*Task: Determination of the stages of IS creation. Stages of sketch design, technical design and development of working documentation and testing) Information system.* *Task for independent work: item 6, number 9.*
Topic 4. IS creation technologies		
10	*Lecture 6. IS creation technologies*	*System requirements. Basic principles. Basic concepts.* *Tasks on SRS: item 6, number 10.*
11	*Lecture 7. Management of the IS development project.*	*Model editing. Sketches. Elements. Step-by-step works.* *Tasks on SRS: item 6, number 11.*
12	*Lecture 8. Planning of the software project*	*Planning of the software project. Function selection and program implementation. Interconnections.* *Task for independent work: item 6, number 12.*
Topic 5. Instrumental means of designing computing systems.		
13	*Lecture 9. Modeling the development of software systems.*	*The task: to master the modeling of the development of software systems. UML language.* *Tasks on SRS: item 6, number 13.*
13	*Lecture 10 Diagrams. Representation. Modeling process.*	*Task: learn Diagrams, Representation, Modeling Process. UML 2.* *Tasks on SRS: item 6, number 13.*
14	*Lecture 11. User interface design. Testing and development of tests.*	*Task: learn the principles of user interface design, testing and test development.* *Task for independent work: item 6, number 14.*
Topic 6. CASE technologies		
15	*Lecture 12 Introduction to CASE technology*	*CASE technologies: what, when, how?* *Task for independent work: item 6, number 15.*
16	*Lecture 13. OVERVIEW OF CASE TOOLS*	*Review and analysis of CASE tools.* *Task for independent work: item 6, number 16.*
Topic 7. Grid and cloud environments for IS design.		
17	*Lecture 14. Introduction to Grid information technologies.*	*Globus Toolkit, ARC, Unicore, gLite and European Middleware Initiative (EMI).* *Task for independent work: item 6, number 17.*
18	*Lecture 15. Designing in cloud environments.*	*Organization of computing in cloud environments. Software as a service. Infrastructure as a service. Platform as a service*

	Windows Azure platform.	*Windows Azure Platform.* *Windows Azure Table.* *Windows Azure Blob.* *Task for independent work: Item 6, No. 18.*
19	*Laboratory work 5 Creating a Windows Azure application*	*Task: practical mastery of creating Windows Azure applications.* *Task for independent work: Item 6, No. 19.*
20	*Laboratory work 6 Work in Windows Azure Table.* *.*	*Task: practical assimilation of work in Windows Azure Table.* *Task for independent work: Item 6, No. 20.*
21	*Lab 7 Work in Windows Azure Blob.*	*Task: practical assimilation of work in Windows Azure Blob.* *Task for independent work: item 6, number 21.*
22	*Lecture 16. Distributed databases*	*Distributed database management system. Design aspects of distributed systems. Fragmentation. Distribution. Replication. Distribution of data.* *Analysis of relational (Postgres), keys and values (Riak, Redis), columnar (HBase), document-oriented (MongoDB, CouchDB), graph databases (Neo4J)* *Task for independent work: item 6, number 22.*
Topic 8 SOLIDWORKS design system.		
23	*Lecture 17. SOLIDWORKS design system.*	*SOLIDWORKS program. Basic principles of SOLIDWORKS. Basic concepts. Document structure. Legend.* *Task for independent work: Item 6, No. 23.*
24	*Lecture 18. User interface. Design process*	*Windows functions. SOLIDWORKS document windows. Function selection and program response. Design process. Design approach. Model editing. Interconnections. Sketches. Elements. Prefab elements. Drawing. Three-dimensional design.* *Task for independent work: Item 6, No. 24.*
Modular test		

6. Student's independent work

The discipline "Parallel and distributed computing" is based on independent preparations for classroom classes on theoretical and practical topics.

№	*The name of the topic that is submitted for independent study*	*Hours of study*	*References*
1	*Preparation for the lecture 1*	*1*	*1, 2, 3, 7 p. 5-16.*
2	*Preparation for the lecture 2*	*1*	*2, 3, 7 p. 16-28.*
3	*Preparation for lab 1*	*2*	*4, p. 5-11.*
4	*Preparation for the lecture 3*	*1*	*2, 3, 7 p. 28-36.*
5	*Preparation for lab 2*	*2*	*4, p. 11-21.*

6	*Preparation for the lecture 4*	*1*	*2,3, 7 p.36-43.*
7	*Preparation for lab 3*	*2*	*4, p. 21-29.*
8	*Preparation for the lecture 5*	*1*	*1, 2, 3, 7 p. 43-53.*
9	*Preparation for lab 4*	*3*	*4, p. 29-33.*
10	*Preparation for the lecture 6*	*1*	*2, 3, 7 p. 53-68.*
11	*Preparation for the lecture 7*	*1*	*2, 3, 7 p. 68-79.*
12	*Preparation for the lecture 8*	*1*	*2, 3, 7 p. 79-88.*
13	*Preparation for the lecture 9*	*1*	*7 p. 88-110; 8*
14	*Preparation for the lecture 10*	*1*	*7 p. 110-130; 8*
15	*Preparation for the lecture 11*	*1*	*7 p. 130-142; 8*
16	*Preparation for the lecture 12*	*1*	*2, 7 p. 142-155;*
17	*Preparation for the lecture 13*	*1*	*2, 7 p. 155-169;*
18	*Preparation for the lecture 14*	*5*	*5, p. 50-64; 7 p. 169-188; 9*
19	*Preparation for the lecture 15*	*3*	*5; 7 p. 188-236; 10*
20	*Preparation for lab 5*	*4*	*4, p. 40-52.*
21	*Preparation for lab 6*	*4*	*4, p. 52-61.*
22	*Preparation for lab 7*	*4*	*4, p. 61-74.*
23	*Preparation for the lecture 16*	*2*	*5, p. 170-214; 10*
24	*Preparation for the lecture 17*	*3*	*7 p. 253-266; 11*
25	*Preparation for the lecture 18*	*5*	*7 p. 266-279; 11*
26	*Preparation for modular test*	*14*	*1; 2; 3; 5 - 11*
27	*Preparation for exam*	*30*	*1; 2; 3; 5 - 11*

Policy and Assessment

7. Політика навчальної дисципліни (освітнього компонента)

- *Attending lectures is mandatory.*
- *Attending laboratory classes can be occasional and as needed to protect the completed laboratory work.*
- *Rules of behavior in classes: activity, respect for those present, turning off phones.*
- *Adherence to the policy of academic integrity.*
- *Rules for the protection of laboratory work: the work must be done according to the student's option, which is determined by his number in the group list.*
- *The rules for assigning incentive and penalty points are as follows.*

Incentive points are awarded for:

- accurate and complete answers in surveys based on lecture materials (maximum number of points for a blitz survey - 3 points).

- creative approach in performing laboratory work (maximum number of points for work - 2 points). Penalty points are calculated for: - plagiarism (inconsistency of the task variant, identity of the program implementation among different works): -5 points for each attempt.

8. Monitoring and grading policy

During the semester, students perform 7 laboratory works. The maximum number of points for each of the laboratory works: 5 points.
Points are awarded for:
- quality of laboratory work (computer workshop): 0-2 points;
- answer during the defense of laboratory work (computer workshop): 0-2 points;
- timely submission of work for defense: 0-1 point.
Performance evaluation criteria:
2 points – the work is done qualitatively, in full;
1 point - the work is completed in full, but contains minor errors;
0 points – the work is incomplete or contains significant errors.
Answer evaluation criteria:
2 points – the answer is complete, well-argued;
1 points – in general, the answer is correct, but has shortcomings or minor errors;
0 points - there is no answer or the answer is incorrect.
Criteria for evaluating the timeliness of work submission for defense:
1 point – the work is submitted for defense no later than the specified deadline;
0 points – the work is submitted for defense later than the specified deadline.
The maximum number of points for performing and defending computer practicals:
5 points × 7 lab. works (comp. practical) = 35 points.
The assignment for the modular test consists of 3 theoretical and 2 practical questions. The answer to each question is evaluated by 3 points.
Evaluation criteria for each test question:
3 points – the answer is correct, complete, well-argued;
2 points - in general, the answer is correct, but has flaws;
1 points – there are significant errors in the answer;
0 points - there is no answer or the answer is incorrect.
The maximum number of points for a modular control work:
3 points × 5 questions = 15 points.
The examination task consists of 3 questions - 2 theoretical and 1 practical The answer to each theory question is worth 15 points, and the answer to a practical question is worth 20 points.
Evaluation criteria for each theoretical question of the exam task:
14-15 points – the answer is correct, complete, well-argued;
11-13 points – the answer is correct, detailed, but not very well argued;
8-10 points - in general, the answer is correct, but has flaws;
5-7 points – there are minor errors in the answer;
1-4 points – there are significant errors in the answer;
0 points - there is no answer or the answer is incorrect.
Evaluation criteria for the practical question of the exam task:
18-20 points - the answer is correct, the calculations are completed in full;
14-17 points - the answer is correct, but not very well supported by calculations;
9-13 points - in general, the answer is correct, but has flaws;

5-8 points – there are minor errors in the answer;

1-4 points – there are significant errors in the answer;

0 points - there is no answer or the answer is incorrect.

The maximum number of points for completing the examination work:

15 points × 2 theory questions + 20 points × 1 theory question = 50 points.

The rating scale for the discipline is equal to: R = Rc = R*lab*+ R*modular control*+ R*exam* = 35 points + 15 points + 50 points = 100 points.

Calendar control: conducted twice a semester as a monitoring of the current status of meeting the syllabus requirements.

At the first certification (8th week), the student receives "credited" if his current rating is at least 10 points (50% of the maximum number of points a student can receive before the first certification).

At the second certification (14th week), the student receives "passed" if his current rating is at least 20 points (50% of the maximum number of points a student can receive before the second certification).

Semester control: exam

Conditions for admission to semester control:

With a semester rating (RC) of not less than 30 points and the admission of all the work of the computer workshop, the student is admitted to the exam. After passing the exam, a grade is assigned according to the table (Table of correspondence of rating points to grades on the university scale).

A necessary condition for admission to the exam is the performance and defense of laboratory work.

Table of correspondence of rating points to grades on the university scale:

Points	*Grade*
100-95	Excellent
94-85	Very good
84-75	Good
74-65	Satisfactorily
64-60	Enough
< 60	Unsatisfactorily
Admission conditions are not met	Not admitted

9. Additional information about the course

The list of questions to be submitted for semester control is given in Appendix 1.

Course syllabus:

Is created by PhD, Associate Professor Vasyl Yurchyshyn

Approved by Computer Systems Software Department (protocol № 8, 22.01.25)

Approved by the Applied Mathematics Faculty Methodology Commission (protocol №8, 03.02.2025)

Appendix 1. List of questions to be submitted for semester control

1. Concept of information system (IS).
2. Structure and hierarchy of information systems. Modular structure of the system.
3. IS classification.
4. Information systems of automated design
5. IS life cycle
6. Pre-project inspection of IS.
7. Creation of an information system.
8. Commissioning of IS.
9. Operation of the information system.
10. IS decommissioning
11. Stages and stages of information system development.
12. Search research work - tasks, goals, stages and stages.
13. Appropriate research work - tasks, goals, stages and stages.
14. Draft project - tasks, goals, stages, composition of documentation.
15. Technical project - tasks, goals, stages, composition of documentation.
16. Development of working documentation of the experimental model.
17. Testing of the prototype at the stage of development of working documentation.
18. The concept and purpose of the test program and test methodology.
19. Departmental, interdepartmental, state, interstate tests of developed systems.
20. Implementation of the developed systems into production and operation.
21. Testing of developed systems at the implementation stage.
22. Author support of developed systems during industrial operation.
23. Responsibilities of the System developer at the stage of operation.
24. Life cycle models of the information system.
25. IS design technology.
26. IS design methods and tools.
27. Standards for design and development of information systems.
28. Types of standards (corporate, industry, state, international).
29. Design documentation standards.
30. User interface standard.
31. Life cycle process standards (LC).
32. Life cycle processes.
33. Interrelationships between the processes of the residential complex.
34. Information technology architecture of IS.
35. "file - server" architecture.
36. Shared ICs (applications and databases).
37 Parallelism in information technologies.

38. Distributed computer systems in the design of modern computer systems.
39. Computing cluster and supercomputer in the design of modern computing systems. Features of programming.
40. Information technologies Grid for building high-performance computing systems.
41. Design of high-performance computing systems based on cloud technologies.
42. Cloud Computing and Grid Computing. Trends of modern infrastructure solutions.
43. Virtualization technologies. Virtualization platforms.
44. General principles of programming in the cloud..
45. Basic principles of SOLIDWORKS.
46. SOLIDWORKS document windows.
47. SOLIDWORKS user interface.
48. Model editing. SOLIDWORKS Relationships.
49. Sketches. SOLIDWORKS elements.
50. Prefab SOLIDWORKS elements.
51. Drawings (drawing documents) SOLIDWORKS.
52. SOLIDWORKS three-dimensional design.