

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ
ТАРАСА ШЕВЧЕНКА
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК ТА КІБЕРНЕТИКИ
Кафедра інформаційних систем**

«ЗАТВЕРДЖУЮ»
Заступник декана
з навчальної роботи
Кашпур О.Ф.
«26» 03 2018 року

**РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
«ПРОДУКТИВНІСТЬ ТА БЕЗПЕКА ПРОГРАМНОГО
ЗАБЕЗПЕЧЕННЯ»
«PERFORMANCE AND SECURITY OF SOFTWARE SYSTEMS»
для здобувачів освітньо-наукового рівня «доктор філософії»
for obtaining an educational-scientific degree: Doctor of Philosophy**

галузь знань	12 «Інформаційні технології»/ “Information Technologies”
спеціальність	121 «Інженерія програмного забезпечення» / “Software Engineering”
освітній рівень	третій (освітньо-науковий)/ third (educational-scientific level)
освітньо-наукова програма	«Інженерія програмного забезпечення» / “Software Engineering”
вид дисципліни	вибіркова / selective


Форма навчання	денна / заочна
Навчальний рік	2018/2019
Рік навчання	2
Кількість кредитів ECTS	4
Мова викладання, навчання та оцінювання	англійська/English
Форма заключного контролю	екзамен/exam

Викладачі: **Жереб Костянтин Анатолійович, к.ф.-м.н.**

Пролонговано: на 2019/2020 н.р. (члор. №) «15» 04 2019 р.
на 2020/2021 н.р. (члор. №) «30» 03 2020 р.

КИЇВ – 2018

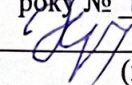
Розробник: **Жеріб Костянтин Анатолійович**, к. ф.-м. н., асистент кафедри інформаційних систем

ЗАТВЕРДЖЕНО Завідувач кафедри
«Інформаційних систем»


(підпис) Проватар О.І.

Протокол № 4 від «21» 12 2017 р.

Схвалено науково-методичною комісією факультету комп'ютерних наук та кібернетики

Протокол від «14» 02 2018 року № 6
Голова науково-методичної комісії  професор, д.ф.-м.н. Хусайнов Д.Я.
(підпис)

1. Метою дисципліни є формування теоретичних знань та практичних умінь в галузі продуктивності та безпеки програмних систем, зокрема використання сучасних моделей, підходів та інструментальних засобів для пошуку недоліків, пов'язаних з продуктивністю та безпекою, підвищення параметрів продуктивності, зокрема часу виконання та використання пам'яті, уникнення розповсюджених вразливостей безпеки.

1. Discipline aim. The aim of the course “Performance and security of software systems” is to form theoretical knowledge and practical skills related to modern aspects of performance and security of software systems, using modern models, approaches and tools to detect issues with performance and security, improve performance metrics such as execution time and memory usage, avoid common security vulnerabilities.

2. Попередні вимоги до опанування або вибору навчальної дисципліни:

- a) *Знати:* принципи організації та архітектуру сучасних програмних систем, мережеві протоколи, веб та мобільну розробку, патерни архітектури та проектування, принципи та методології програмної інженерії, сучасні мови програмування, бібліотеки та фреймворки.
- b) *Вміти:* аналізувати та формалізувати вимоги, проектувати, розробляти та тестувати програмні системи з використанням сучасних мов програмування, бібліотек та фреймворків.

2. Preliminary demands to master or choice of the course discipline

a. *To know:* organization principles and architecture of modern software systems, network protocols, web and mobile development, architecture and design patterns, software engineering principles and methodologies, modern programming languages, libraries and frameworks.

b. *To be able to:* analyze and specify requirements, design, implement, test and verify software systems using modern programming languages, libraries and frameworks.

3. Анотація навчальної дисципліни:

Дисципліна «Продуктивність та безпека програмного забезпечення» належить до переліку дисциплін вільного вибору. Вона забезпечує професійний розвиток, спрямований на формування концептуальних та методологічних знань у галузі інформаційних технологій, вміння критично аналізувати, оцінювати і синтезувати нові та комплексні ідеї, аналізувати складні програмні системи, вміння здійснювати раціональний вибір підходів проектування, побудови та підтримки програмних систем. В рамках дисципліни вивчаються основні принципи та методи побудови надійних та ефективних програмних систем з використанням сучасних підходів та інструментальних засобів.

3. Synopsis of the course:

The course “Performance and security of software systems” belongs to free choice disciplines. It provides professional progress directed on formation of conceptual and methodological knowledge in area of informational technologies, ability to analyze, estimate and synthesize new and complex ideas, analysis of sophisticated software systems and applications, making rational choices and decisions during design, development and maintenance of software systems. Within a frame of this discipline, there is a study of basic principles and methods of building secure and efficient software systems using modern approaches and tools.

4. Завдання (навчальні цілі): набуття знань, умінь та навичок (компетентностей) на рівні новітніх досягнень у методах створення програмного забезпечення, програмуванні, відповідно до науково-освітньої кваліфікації «Доктор філософії». Зокрема, розвивати:

- здатність до абстрактного мислення, аналізу та синтезу (ЗК 1);
- здатність працювати в міжнародному контексті (ЗК 3);
- вміння виявляти, ставити і вирішувати проблеми в галузі інформаційних технологій (ЗК 6);

- здатність накопичувати, обробляти та систематизувати професійні знання в інформаційних технологіях і визнавати важливість навчання протягом всього життя (ФК 2);
- здатність критично переосмислювати наявні інформаційні технології та відстежувати тенденції їх розвитку (ФК 6);
- здатність розвивати і реалізовувати нові конкурентоспроможні ідеї в інженерії програмного забезпечення (ФК 9).

4. Objectives of study:

to obtain the knowledge, skills and the newest level competences in the solution of the problems on level of modern achievement in developing of programs corresponding to the PhD qualification level. In particular, to develop:

- the ability to think abstractly, analyze and synthesize;
- the ability to work in an international context;
- the ability to identify, pose and solve problems in the field of information technology;
- the ability to accumulate, process and systematize professional knowledge in information technology and to recognize the importance of lifelong learning;
- the ability to critically re-think existing information technologies and track trends in their development;
- the ability to develop and implement new competitive ideas in software engineering.

5. Results of learning /Результати навчання за дисципліною:

Результат навчання (РН) (1. знати; 2. вміти; 3. комунікація; 4. автономність та відповідальність)		Форми (та/або методи і технології) викладання і навчання	Методи оцінювання та пороговий критерій оцінювання (за необхідності)	Відсоток у підсумковій оцінці з дисципліни
Код	Результат навчання			
РН 1.1	To know formal and empirical models of performance and security for modern software systems.	<i>Lectures, Individual work</i>	<i>Exam, active participation in lectures, oral answers.</i>	20%
РН 1.2	To know methods, approaches and tools used to measure performance of software systems.			
РН 1.3	To know methods and approaches of improving performance of software systems using modern programming languages, libraries and frameworks.			20%
РН 1.4	To know common security vulnerabilities of modern software systems and methods and approaches to avoid or mitigate these vulnerabilities.			
РН 2.1	To be able to design, develop and deploy efficient and secure software systems	<i>Lectures, Practical work, Individual work</i>	<i>Exam, defense of project, active participation in lectures, oral answers, accomplishment of tasks assigned to Individual work. review defense</i>	20%
РН 2.2	To be able to identify and fix the issues with performance and security of software systems			20%
РН 2.3	To be able to consider trade-offs between performance security and other functional and non-functional requirements and make rational decisions that achieve practical balance between these requirements.	<i>Lectures, Practical work, Individual work</i>	<i>Review defense</i>	5%
РН3.1	To develop own point of view on problem, communication with colleagues on questions of developing, to create writing reports.			5%

PH4.1	Demonstration of authority, innovation, high level individuality, academic and professional honest, consequence devoting of developing new ideas or processes in modern context professional and science activity.			5%
PH4.2	Responsible regards to works, to take responsibility for their quality.			5%

6. Correspondence between learning results and program study results / Співвідношення результатів навчання дисципліни із програмними результатами навчання

Програмні результати навчання	Результати навчання дисципліни									
	PH 1.1	PH 1.2	PH 1.3	PH 1.4	PH 2.1	PH 2.2	PH 2.3	PH 3.1	PH 4.1	PH 4.2
<i>(з опису освітньої програми)</i>										
ПРН-7. Знати, розуміти і самостійно застосовувати методи аналізу предметної області, виявлення інформаційних потреб і збір даних для проектування / ПРН-7. Know, understand and independently apply methods of domain analysis, identification of information needs and data collection for design.	+	+		+	+			+	+	
ПРН-8. Оцінювати, класифікувати і обґрунтовувати вибір методів формування вимог до інформаційної системи, формулювати вимоги / ПРН-8. Evaluate, classify and justify the choice of methods for forming requirements for the information system, formulate requirements.	+	+			+		+	+		
ПРН-9. Аналізувати, оцінювати і вибирати сучасні інструментальні та обчислювальні засоби, технології, алгоритмічні і програмні рішення для конкретної задачі в галузі комп'ютерних наук та інформаційних технологій/ ПРН-9. Analyze, evaluate and select state-of-the-art tools and computing tools, technologies, algorithmic and software solutions for a specific task in the field of computer science and information technology.				+	+	+	+	+		+

7. Схема формування оцінки.

7.1. Форми оцінювання здобувачів освітньо-наукового ступеня:

- оцінювання впродовж навчального періоду:

1. Активна робота на лекції, усні відповіді: PH1.1, PH1.2, PH1.3, PH1.4, PH 2.1, PH 2.2 – 10 балів/6 балів;
2. Виконання завдань, винесених на самостійну роботу: PH2.1, PH2.2 – 10 балів/6 балів;
3. Захист проекту: PH2.1, PH2.2, PH2.3, PH3.1, PH4.1, PH4.2, – 40 балів/24 бали;

- підсумкове оцінювання: екзамен.

- максимальна кількість балів які можуть бути отримані: 40 балів;
- результати навчання які будуть оцінюватись: PH1.1, PH1.2, PH1.3, PH1.4, PH 2.1, PH 2.2;
- форма проведення і види завдань: письмова робота.

Для здобувачів освітньо-наукового ступеня, які набрали сумарно меншу кількість балів ніж критично-розрахунковий мінімум – 20 балів для одержання іспиту за рішенням кафедри не допустити

до складання іспиту із рекомендацією здати контрольні роботи та захистити проект до повторного складання іспиту.

Рекомендований мінімум – 36 балів.

7.2. Організація оцінювання:

Обов'язковим є виконання завдань, винесених на самостійну роботу, та реалізація проекту.

У частину 1 входять теми 1 - 3, у частину 2 – теми 4 – 6 у частину 3 – теми 7 – 9. Обов'язковим для екзамену є виконання завдань, винесених на самостійну роботу та захист проекту до вказаної викладачем дати, перед початком екзаменаційної сесії, згідно навчального плану. Переписування чи перескладання тем не практикується.

Терміни проведення форм оцінювання:

1. Завдання на самостійну роботу, частина 1: до 5 тижня навчального періоду.
2. Завдання на самостійну роботу, частина 2: до 10 тижня навчального періоду.
3. Захист проекту: до 8 тижня навчального періоду.

У випадку відсутності з поважних причин відпрацювання та перездачі завдань здійснюються у відповідності до „Положення про організацію освітнього процесу”.

7.3. Шкала відповідності оцінок

Відмінно / Excellent	90-100
Добре / Good	75-89
Задовільно / Satisfactory	60-74
Незадовільно / Fail	0-59

При визначенні оцінки визначальною є робота протягом навчального періоду. Після завершення розгляду тем проводиться теоретичне опитування.

8. STRUCTURE OF THE DISCIPLINE. THEMATIC PLAN OF LECTURES AND TUTORIALS (СТРУКТУРА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ. ТЕМАТИЧНИЙ ПЛАН ЛЕКЦІЙ І ПРАКТИЧНИХ ЗАНЯТЬ

№	Lecture title (Назва лекції)	Кількість годин		
		Лекції Lecture	Практичн і Tutorials	Individual work
Part 1 (Частина 1). „Performance of software systems”				
1	Theme 1. Introduction. Principles and models of software systems performance. Measuring performance of software systems. <i>Individual work:</i> Measure performance of various software systems, including those used in dissertation work.	2		8
2	Theme 2. Improving performance on source code level. <i>Individual work:</i> Find and eliminate common performance issues in source code of open source projects and/or dissertation work.	2		12
3	Theme 3. Performance of network applications. Performance aspects on server, client and during network transfer. <i>Individual work:</i> Measure performance of different components of client-server application. Investigate performance trade-offs	2	2	8

	when switching to a different architecture (e.g. peer-to-peer)			
4	Theme 4. Performance of data processing. <i>Individual work:</i> Measure performance of database application. Compare performance of different database solutions (relational/SQL, NoSQL, Big Data).	2		12
5	Theme 5. Scalability of software systems. <i>Individual work:</i> Compare performance and scalability requirements. Investigate tools for improving scalability aspects of software systems.	2		8
Part 2 (Частина 2). „Security of software systems”				
6	Theme 6. Principles and models of software systems security. <i>Individual work:</i> Show applications of security principles in open source projects and/or dissertation work.	2		12
7	Theme 7. Common vulnerabilities in modern software systems and approaches to avoid and mitigate these vulnerabilities. <i>Individual work:</i> Use different automated tools for detecting common vulnerabilities. Compare their effectiveness and precision.	2	2	12
8	Theme 8. Principles of improving security of applications against possible future vulnerabilities. <i>Individual work:</i> Investigate well-known security vulnerabilities; determine if they could have been prevented by applying security principles and best practices.	2		12
9	Theme 9. Designing software to meet performance and security requirements. Balancing performance, security and other functional and non-functional requirements. <i>Individual work:</i> Describe trade-offs between performance, security and other requirements using examples from open-source projects and/or dissertation work.	2		12
TOTAL		18	4	96

Total duration/Загальний обсяг 120 годин, в тому числі:

Лекцій – **18 годин**,

Практичні – **4 години**.

Консультації - **2 години**.

Самостійна робота – **96 годин**.

9. Recommended sources /Рекомендовані джерела

Main/Основні:

1. Gregg, Brendan. Systems Performance: Enterprise and the Cloud. – Prentice Hall, 2013. – 792 p.
2. Armstrong, Tavish (ed.). The Performance of Open Source Applications. – <https://www.aosabook.org/en/posa/introduction.html>

3. Kunkas, Jim. Power and Performance: Software Analysis and Optimization. – Morgan Kaufmann, 2015. – 300 p.
4. Liu, Henry H. Software performance and scalability: a quantitative approach. – John Wiley & Sons, 2011. – 375 p.
5. Bondi, André B. Foundations of software and system performance engineering: process, performance modeling, requirements, testing, scalability, and practice. – Pearson Education, 2015. – 446 p.
6. Ransome, James, and Anmol Misra. Core software security: Security at the source. – CRC press, 2018. – 416 p.
7. Peltier, Thomas R. Information Security Fundamentals, Second Edition. – Auerbach Publications, 2013. – 438 p.
8. Open Web Application Security Project (OWASP) – <https://owasp.org/>

Additional/Додаткові:

9. Molyneaux, Ian. The art of application performance testing: from strategy to tools. – O'Reilly Media, Inc., 2014. – 278 p.
10. Gerber, Richard, et al. The software optimization cookbook: high-performance recipes for IA-32 Platforms. – Intel Press, 2005. – 400 p.
11. Sehr, Viktor, Bjorn Andrist. C++ High Performance: Boost and optimize the performance of your C++ 17 code. – Packt Publishing, 2018. – 374 p.
12. Gorelick, Micha, and Ian Ozsvald. High Performance Python: Practical Performant Programming for Humans. – O'Reilly Media, Inc., 2014. – 370 p.
13. Adams, Chad R. Mastering JavaScript High Performance. – Packt Publishing, 2015. – 208 p.
14. Theron, Vincent, and Michael Diamant. Scala High Performance Programming. – Packt Publishing Ltd, 2016. – 274 p.
15. P.I. Andon, A.Yu. Doroshenko, K.A. Zhreb, O.A. Yatsenko. Algebra-Algorithmic Models and Methods of Parallel Programming. – Kyiv, "Akademperiodyka".–2018. – 192 p.
16. Takanen, Ari et al. Fuzzing for Software Security Testing and Quality Assurance. – Artech House Publishers, 2018. – 330 p.
17. Howard, Michael. 24 Deadly Sins of Software Security: Programming Flaws and How to Fix Them. – McGraw-Hill Education, 2010. – 432 p.
18. Dowd, Mark. The Art of Software Security Assessment: Identifying and Preventing Software Vulnerabilities. – Addison-Wesley Professional, 2007. – 1200 p.
19. McGraw, Gary. Software Security: Building Security In. – Addison-Wesley Professional, 2006. – 448 p.