

АЛГОРИТМІЗАЦІЯ ТА ПРОГРАМУВАННЯ

СИЛАБУС

Шифр і назва спеціальності	122 — Комп'ютерні науки	Інститут / факультет	Комп'ютерних наук і програмної інженерії
Назва програми	«Комп'ютерні науки та інтелектуальні системи»	Кафедра	програмної інженерії та інформаційних технологій управління
Тип програми	Освітньо-професійна	Мова навчання	Українська

Викладач

ПІБ, електронна пошта	Білова Марія Олексіївна, mariia.bilova@khp.edu.ua
-----------------------	--



к.т.н. доцент кафедри програмної інженерії та інформаційних технологій управління. Підготувала і опублікувала понад 50 публікацій, 2 колективні монографії, 1 підручник з грифом університету, 3 статей у виданнях, індексованих у Scopus. (h-index = 3, i10-index = 0 у Google Академії-<https://scholar.google.com/citations?user=b3YLGToAAAAJ>; ідентифікатор ORCID-<https://orcid.org/0000-0001-7002-4698>, ідентифікатор автора Scopus <https://www.scopus.com/authid/detail.uri?authorId=57190442390>).

Провідний лектор з курсів: *Алгоритмізація та програмування (частина 1, частина 2) (українською мовою)*

Загальна інформація про курс

Анотація	Дисципліна «АЛГОРИТМІЗАЦІЯ ТА ПРОГРАМУВАННЯ» є навчальною дисципліною з циклу професійної обов'язкової підготовки за спеціальністю 122 «Комп'ютерні науки». Вона викладається у першому та другому семестрах в обсязі 300 год.(10 кредитів ECTS), зокрема: лекції – 80 год., лабораторні – 80 год., самостійна робота – 140 год. У курсі передбачено чотири змістових модулі та п'ять модульних контрольних робіт. Завершуються обидві частини дисципліни екзаменом.
Цілі курсу	Формування у студентів алгоритмічного мислення та набуття навичок розробки програм на мовах програмування C та Python для вирішенні прикладних задач із різних предметних областей відповідно до сучасних принципів і тенденцій побудови

	прикладного програмного забезпечення.						
Формат	Лекції, лабораторні роботи, самостійна робота. Підсумковий контроль — іспит.						
Семестр	1, 2						
Обсяг (кредити) / Тип курсу (обов'язковий / вибірковий)	10 / Обов'язковий	Лекції (години)	80	Лекційні заняття (години)	80	Самостійна робота (години)	140

Програмні компетенції	<p>ЗК1. Здатність до абстрактного мислення, аналізу та синтезу.</p> <p>ЗК2. Здатність застосовувати знання у практичних ситуаціях.</p> <p>ЗК3. Знання та розуміння предметної області та розуміння професійної діяльності.</p> <p>ЗК6. Здатність вчитися й оволодівати сучасними знаннями.</p> <p>ЗК7. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.</p> <p>ЗК9. Здатність працювати в команді.</p> <p>СК3. Здатність до логічного мислення, побудови логічних висновків, використання формальних мов і моделей алгоритмічних обчислень, проектування, розроблення й аналізу алгоритмів, оцінювання їх ефективності та складності, розв'язності та нерозв'язності алгоритмічних проблем для адекватного моделювання предметних областей і створення програмних та інформаційних систем.</p> <p>СК8. Здатність проектувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: узагальненого, об'єктно-орієнтованого, функціонального, логічного, з відповідними моделями, методами й алгоритмами обчислень, структурами даних і механізмами управління.</p>
------------------------------	---

Результати навчання	Методи викладання та навчання	Форми оцінювання (поточне оцінювання CAS, підсумкове оцінювання FAS)
<p>ПР1. Застосовувати знання основних форм і законів абстрактно-логічного мислення, основ методології наукового пізнання, форм і методів вилучення, аналізу,</p>	<p>Інтерактивні лекції з презентаціями, дискусії, лабораторні заняття, командна робота, кейс-метод, метод зворотного зв'язку з боку студентів, проблемне навчання</p>	<p>Письмові індивідуальні завдання до лабораторних робіт (CAS), оцінювання знань на лабораторних заняттях (CAS), експрес-опитування (CAS), онлайн -тести (CAS), підсумковий/семестровий контроль у формі семестрового екзамену, відповідно до графіку</p>

обробки та синтезу інформації в предметній області комп'ютерних наук.		навчального процесу (FAS)
ПР5. Проектувати, розробляти та аналізувати алгоритми розв'язання обчислювальних та логічних задач, оцінювати ефективність та складність алгоритмів на основі застосування формальних моделей алгоритмів та обчислюваних функцій.	Інтерактивні лекції з презентаціями, дискусії, лабораторні заняття, командна робота, кейс-метод, метод зворотного зв'язку з боку студентів, проблемне навчання	Письмові індивідуальні завдання до лабораторних робіт (CAS), оцінювання знань на лабораторних заняттях (CAS), онлайн -тести (CAS), підсумковий/семестровий контроль у формі семестрового екзамену, відповідно до графіку навчального процесу (FAS)
ПР9. Розробляти програмні моделі предметних середовищ, вибирати парадигму програмування з позицій зручності та якості застосування для реалізації методів та алгоритмів розв'язання задач в галузі комп'ютерних наук.	Інтерактивні лекції з презентаціями, дискусії, лабораторні заняття, командна робота, кейс-метод, дослідження, проєктне навчання	Письмові індивідуальні завдання до лабораторних робіт (CAS), оцінювання знань на лабораторних заняттях (CAS), онлайн -тести (CAS), підсумковий/семестровий контроль у формі семестрового екзамену, відповідно до графіку навчального процесу (FAS)

СИСТЕМА ОЦІНЮВАННЯ

Критерії оцінювання успішності студента	Сума балів за всі види навчальної діяльності	Оцінка ECTS	Оцінка за національною шкалою	Нарахування балів	100% підсумкове оцінювання у вигляді іспиту (30%) та поточного оцінювання (70%). 30% іспит:
	90-100	A	відмінно		
	82-89	B	добре		
	74-81	C			
	64-73	D	задовільно		
	60-63	E			

	35-59	FX	незадовільно з можливістю повторного складання		семестровий екзамен, відповідно до графіку навчального процесу 70% поточне оцінювання:
	0-34	F	незадовільно з обов'язковим повторним вивченням дисципліни		<ul style="list-style-type: none"> • 30-45% оцінювання завдань на лабораторних роботах; • 25-40% проміжний контроль (модульні контрольні роботи та індивідуальні домашні завдання);

Політика курсу

Студенти зобов'язані відвідувати заняття згідно розкладу та дотримуватися етики поведінки. У разі відсутності студентам необхідно буде виконати всі завдання, щоб компенсувати пропущені заняття. Участь у лабораторних заняттях вимагає попередньої підготовки та завчасного опрацювання всіх необхідних матеріалів для продуктивних дискусій під час заняття. Письмові завдання повинні бути подані до встановлених строків.

Структура та зміст курсу

Семестр 1

Тема 1	Алгоритми особливості розробки та їх	Лабораторна робота 0	Проект Scratch		Самостійна робота	Розробка діаграми діяльності відповідно до завдання
Тема 2	Системи числення	Лабораторна робота 1	Основні синтаксичні конструкції мови програмування С			Перехід між системами числення
Тема 3	Типи даних та змінні в мові С	Лабораторна робота 2	Масиви та функції			Реалізація програми на основі розробленої раніше діаграми діяльності
Тема 4	Умовні оператори	Лабораторна робота 3	Алгоритми			Реалізація сортування масиву двома алгоритмами на вибір викладача
Тема 5	Цикли	Лабораторна робота 4	Робота з пам'яттю			Реалізація структури відповідно до завдання
Тема 6	Аналіз працездатності програми	Лабораторна робота 5	Структури даних			
Тема 7	Масиви					
Тема 8	Функції					
Тема 9	Сортування					
Тема 10	Особливості роботи з пам'яттю					
Тема 11	Структури					

Семестр 2						
Тема 1	Мова програмування Python. Типи даних	Лабораторна робота 1	Основні концепції мови програмування Python		С а м о с т і й н а р о	Діаграма класів. Діаграма об'єктів. Діаграма пакетів
Тема 2	Умови та цикли у Python	Лабораторна робота 2	Основи SQL			Особливості роботи з CSS властивостями при розробці сайту.
Тема 3	Функції та особливості роботи з рядками у Python	Лабораторна робота 3	Основи HTML та CSS			
Тема 4	Тема 4. Основи мови SQL	Лабораторна робота 4	Створення застосунку на основі Flask.			
Тема 5	Основи UML					

Тема 6	Основи веб-розробки				б	
Тема 7	Основи backend розробки				о	
					т	
					а	

Література

1. Корнієнко, М. М., Іванова, І. Д. (2011). Інформатика. Основи алгоритмізації і програмування. Ранок.
2. Cormen, T. H. Thomas, H. Cormen Charles E., Leiserson Ronald L., Rivest Clifford Stein. (2009). Introduction to Algorithms Third Edition. The MIT Press.
3. Stephens R. (2013). Essential Algorithms: A Practical Approach to Computer Algorithms. John Wiley & Sons, Inc.
4. Pearson Education. (2014). C Programming Absolute Beginner's Guide. (3rd ed.).
5. Seacord R. C. (2020). Effective C: An Introduction to Professional C Programming. No Starch Press.
6. Thomas Mailund. Pointers in C Programming. A Modern Approach to Memory Management, Recursive Data Structures, Strings, and Arrays. Apress.
7. Anquetil, R. (2019). Fundamental Concepts for Web Development: HTML5, CSS3, JavaScript and much more. Independently published.
8. Myers M. (2017). Smarter way to learn Python.
9. Stephenson, B. (2019). The Python Workbook. Springer: Texts in Computer Science.
10. Маттес, Е. (2021). Пришвидшений курс Python. Видавництво Старого Лева.
11. Руденко, В. Д., Жугастров, О. О. (2019). Основи алгоритмізації і програмування мовою Python. Ранок.
12. Allen, G. (2019). Taylor Author of SQL All-in-One For Dummies. (9th ed.). Hoboken.
13. Upadhyay, K. Ch. (2020). HTML5 For Web Designers. Complete Hypertext Markup Language Guidance. Independently published.
14. Grant, K. J. CSS in depth. Manning Publications Co, 2018. – 445 p.
15. Grinberg M. Flask Web Development: Practical Patterns and Recipes. (2018). O'Reilly.
16. Skiena, S. S. (2020). The Algorithm Design Manual. (3rd. ed.). Springer, Texts in Computer Science.
17. No Starch Press. (2017). Al Sweigart Invent Your Own Computer Games with Python. (4th ed.).
18. Мартін, Р. С. (2019). Чистий код.
19. Chacon, S. Ben Straub. (2014). Apress. Retrieved from <https://git-scm.com/book/uk/v2>.
20. The GNU C Reference Manual. Retrieved from <https://www.gnu.org/software/gnu-c-manual/gnu-c-manual.html>
21. Beej's Guide to C Programming. Retrieved from <https://beej.us/guide/bgc/html//index.html>
22. CS50 «Основи програмування» на платформі Prometheus. Відновлено з https://edx.prometheus.org.ua/courses/Prometheus/CS50/2016_T1/info
23. CS50 Retrieved from <https://cs50.harvard.edu/>
24. C Programming Language Documentation. Retrieved from <https://devdocs.io/c/>
25. Cody Lindley (2018). Front-End Developer Handbook 2018. Frontend Masters. Retrieved from <https://legacy.gitbook.com/book/frontendmasters/front-end-developer-handbook-2018/details>.
26. SQL Tutorial. Retrieved from. <https://www.w3schools.com/sql/>
27. CSS Snapshot 2017. W3C Working Group Note Retrieved from <https://www.w3.org/TR/css-2017/>
28. HTML 5.2. W3C Recommendation. Retrieved from <https://www.w3.org/TR/html52/>.
29. UML. Retrieved from. <https://www.uml.org/>
30. Основи UML. Retrieved from <https://docs.kde.org/trunk5/uk/umbrello/umbrello/uml-basics.html>

Норми академічної етики

Студент повинен дотримуватися «Кодексу етики академічних взаємовідносин та доброчесності НТУ «ХПІ»:
виявляти дисциплінованість, вихованість, доброзичливість, чесність, відповідальність

Силабус за змістом повністю відповідає робочій програмі курсу.