

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Львівський національний університет імені Івана Франка
Факультет прикладної математики та інформатики
Кафедра програмування

Затверджено

На засіданні кафедри програмування
факультету прикладної математики
Львівського національного університету
імені Івана Франка
(протокол № 1 від 29 серпня 2023 р.)



Зав. кафедри к. ф.-м. н., доц. Ярошко С. А.

Силабус навчальної дисципліни
«Алгоритмічні моделі інформатики»,
викладається в межах ОПП «Середня освіта (Інформатика)»
другого (магістерського) рівня вищої освіти
за предметною спеціальністю 014.09 Середня освіта (Інформатика)
спеціальності 014 – Середня освіта,
галузі знань 01 – Освіта/Педагогіка

Львів 2023 р.

Назва дисципліни	Алгоритмічні моделі інформатики
Адреса викладання дисципліни	Львівський національний університет імені Івана Франка, вул. Університетська 1, м. Львів, Україна, 79000
Факультет та кафедра, за якою закріплена дисципліна	Факультет прикладної математики та інформатики, кафедра програмування
Галузь знань, шифр та назва спеціальності	Галузь знань: 01 Освіта / Педагогіка Спеціальність: 014 Середня освіта Спеціалізація: 014.09 Середня освіта (Інформатика)
Викладачі дисципліни	Черняхівський Володимир Вікторович, к. ф.-м. н., доц., доцент кафедри програмування Селіверстов Роман Григорович, к. ф.-м. н., доц., доцент кафедри програмування
Контактна інформація викладачів	<i>E-mail:</i> volodymyr.chernyakhivskyy@lnu.edu.ua roman.seliverstov@lnu.edu.ua <i>Веб-сторінки:</i> https://ami.lnu.edu.ua/employee/cherniakhivskyy https://ami.lnu.edu.ua/employee/seliverstov-roman-hryhorovych
Консультації з питань навчання по дисципліні відбуваються	Консультації проводять раз на тиждень згідно з оприлюдненим розкладом консультацій викладача. Можливі онлайн консультації через Zoom чи Microsoft Teams. Для погодження часу онлайн консультацій потрібно писати на електронну пошту викладача.
Сторінка курсу	https://ami.lnu.edu.ua/course/alhorytmichni-modeli-informatyky-so стаціонар https://ami.lnu.edu.ua/course/alhorytmichni-modeli-informatyky-so-zfn заочне
Інформація про дисципліну	Дисципліна “Алгоритмічні моделі інформатики” є нормативною дисципліною з предметної спеціальності середня освіта (інформатика) для освітньої програми "Середня освіта (Інформатика)". Дисципліну викладають: <u>для стаціонарного навчання</u> в першому семестрі магістерського рівня в обсязі 6 кредитів (за Європейською Кредитно-Трансферною Системою ECTS); <u>для заочного навчання</u> в другому семестрі магістерського рівня в обсязі 6 кредитів. За результатами вивчення дисципліни студенти складають іспит.
Коротка анотація дисципліни	Курс розроблено таким чином, щоб надати студентам необхідні знання, обов'язкові для викладання розділів інформатики за програмою середньої загальноосвітньої школи в частині алгоритмізації. Розглядають різні аспекти підготовки для здобуття кваліфікації магістра: алгоритми для початківців; виконавці алгоритмів та їх системи команд; моделі предметних областей для цілей алгоритмізації; алгоритми як метод опрацювання інформації; письмові, графічні, математичні, програмні методи розробки і реалізації алгоритмів. Крім того, здобувачі отримують розширене коло знань про алгоритмічні моделі на рівні вищої освіти: обернені задачі; семантичні моделі програмування; скінченні автомати; алгоритмічні мови; інформаційні моделі. Курс використовує низку знань і навиків, які були отримані студентами на курсах бакалаврату за розділами програмування і суміжних з ними.
Мета та цілі дисципліни	Метою нормативної дисципліни «Алгоритмічні моделі інформатики» є: <ul style="list-style-type: none"> • формування базової системи знань та навиків для викладання розділів інформатики за програмою середньої загальноосвітньої школи в частині алгоритмізації; • отримання навиків самостійної розробки моделей алгоритмізації, потрібних для різних прикладних цілей і для побудови навчальних планів учнів; • формування алгоритмічного мислення;

	<ul style="list-style-type: none"> ● набуття компетенцій, знань, умінь та навиків і прийомів з викладання інформатики у середній школі; ● вивчення методів організації роботи учнів школи для цілей алгоритмізації; ● вивчення сучасних прийомів формування теоретичних знань і практичних навичок учнів.
<p>Література для вивчення дисципліни</p>	<p><i>Основна література</i></p> <ol style="list-style-type: none"> 1. Підручники інформатики для середніх загальноосвітніх шкіл. 2. МійКлас. Електронний освітній ресурс. – Режим доступу: https://miyklas.com.ua/ 3. Програмування по українськи. Сайт різноманітних матеріалів до вивчення програмування [Електронний ресурс]. – Режим доступу: http://programming.in.ua/ 4. Вікіпедія. Алгоритм [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Алгоритм 5. Вікіпедія. Список алгоритмів [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Список_алгоритмів 6. Blockly. Бібліотека JavaScript для створення візуальних редакторів програмування (візуальний редактор коду) [Електронний ресурс]. – Режим доступу: https://developers.google.com/blockly/ 7. turtle – графіка черепахи (модуль Python) [Електронний ресурс]. – Режим доступу: https://docs.python.org/3/library/turtle.html 8. Real Python. Посібник для початківців з Python Turtle [Електронний ресурс]. – Режим доступу: https://realpython.com/beginners-guide-python-turtle/ 9. Algovis - добірка ресурсів з візуалізацією алгоритмів [Електронний ресурс]. – Режим доступу: https://proglib.io/p/visualising-algorithms/ 10. Visu Algo. Візуалізація структур даних і алгоритмів за допомогою анімації [Електронний ресурс]. – Режим доступу: https://visualgo.net/en 11. Візуалізація структури даних. Університет комп'ютерних наук в Сан-Франциско [Електронний ресурс]. – Режим доступу: https://www.cs.usfca.edu/~galles/visualization/Algorithms.html 12. Вікіпедія. Синтаксис (програмування) [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Синтаксис_(програмування) 13. Вікіпедія. Семантика мов програмування [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Семантика_мов_програмування 14. Вікіпедія. Скінченний автомат [Електронний ресурс]. – Режим доступу: https://uk.wikipedia.org/wiki/Скінченний_автомат 15. Всеукраїнські олімпіади з інформатики. Сайт [Електронний ресурс]. – Режим доступу: https://oi.in.ua/ <p><i>Додаткова література</i></p> <ol style="list-style-type: none"> 16. Cormen T.H. Introduction To Algorithms / Cormen, Thomas H.; Leiserson, Charles E.; Rivest, Ronald L.; Stein, Clifford // MIT Press. – 2001. – 1180 с. – Режим доступу: https://books.google.com.ua/books?id=NLNgYyWFl_YC&redir_esc=y 17. Костів О.В. Методи розробки алгоритмів: Тексти лекцій / О. В. Костів, С. А. Ярошко // Львів. нац. ун-т ім. І.Франка. – Львів, 2002. – 99 с. 18. Прийма С. М. Теорія алгоритмів: Навчальний посібник / С.М.Прийма // Мелітополь: ФОП Однорог Т. В. – 2018. – 116 с.
<p>Обсяг курсу</p>	<p>6 кредитів ЄКТС – 180 годин. З них: <u>стаціонарне навчання</u> – 32 години лекцій, 32 години лабораторних занять та 116 годин самостійної роботи; <u>заочне навчання</u> – 12 годин лекцій, 12 годин лабораторних занять та 156 годин самостійної роботи.</p>
<p>Очікувані результати навчання</p>	<p>Після завершення цього курсу студент буде:</p> <p><i>знати</i></p> <ul style="list-style-type: none"> ● предмет, методи та завдання дисципліни; ● прийоми викладання розділів інформатики в СЗШ в частині алгоритмізації;

	<ul style="list-style-type: none"> ● універсальні і формальні алгоритмічні моделі; ● моделі алгоритмів для різних вікових груп учнів; ● процедури аналізу, проектування і розробки алгоритмів за різними системами команд і виконавців; ● письмові, графічні, математичні, програмні методи розробки і реалізації алгоритмів; <p><i>вміти</i></p> <ul style="list-style-type: none"> ● обирати правильну модель вивчення алгоритмів для учнів різних вікових категорій; ● будувати уроки інформатики різного типу для розділів алгоритмізації; ● застосовувати до конкретних прикладних задач відповідну алгоритмічну модель; ● формально, словесно, графічно, з використанням псевдокоду та мови програмування високого рівня записувати алгоритми; ● реалізовувати і досліджувати обчислювальні, графічні та інші алгоритми засобами середовищ Blockly, turtle, Algovis, algorithm-visualizer тощо; ● обирати і застосовувати різні методи контролю знань учнів.
<p>Компетентності</p>	<p><i>Інтегральна:</i> Здатність застосовувати фундаментальні та поглиблені теоретичні знання в освіті та для розв'язування складних спеціалізованих задач та практичних проблем у галузі комп'ютерних наук або у процесі навчання, що передбачає застосування теорій та методів комп'ютерних наук, інформаційних технологій і характеризується комплексністю та невизначеністю умов.</p> <p><i>Загальні (ЗК):</i></p> <p>ЗК1. Здатність до абстрактного мислення, аналізу та синтезу. ЗК2. Знання та розуміння предметної області та розуміння професії. ЗК3. Здатність спілкуватися рідною мовою як усно, так письмово. ЗК5. Навички використання інформаційних і комунікаційних технологій при навчанні школярів та проведенні досліджень на відповідному рівні. ЗК6. Здатність до пошуку, опрацювання та аналізу інформації з різних джерел для організації навчального процесу. ЗК7. Здатність виявляти, ставити та вирішувати проблеми. ЗК8. Здатність приймати обґрунтовані рішення та працювати в команді.</p> <p><i>Фахові компетентності спеціальності (ФК):</i></p> <p>ФК1. Здатність публічно висловлювати та аргументувати наукові твердження, подавати результати досліджень, аналізувати результати інших авторів у вибраному напрямі. ФК3. Здатність застосовувати набуті педагогічні знання та знання з психології в навчальному процесі. ФК4. Здатність аналізувати основні положення концепцій, методологічних пріоритетів, соціально-ієрархічних зв'язків у системі пізнання. ФК5. Здатність використовувати граматичні, стилістичні та дискурсивні аспекти іноземної мови науково-професійного спрямування в області комп'ютерних наук. ФК8. Здатність вивчати нові методики викладання навчальних дисциплін як теоретичного так і практичного напрямку, виробничих та педагогічних практик, курсового та дипломного проектування, самостійної роботи. ФК9. Здатність обґрунтовувати теоретико-філософські засади організації навчального процесу; визначати головні функції, обов'язки вчителя. ФК10. Здатність проектувати, будувати та налаштовувати системи навчання, в процесі яких учні здобувають знання шляхом планування і виконання практичних завдань (проектів), які поступово ускладнюються. ФК12. Здатність до використання та проектування методик навчання учнів загальноосвітніх та спеціалізованих шкіл інформаційним технологіям. ФК13. Сучасний світогляд на інформацію і її роль в інформаційному суспільстві. ФК17. Здатність використовувати процедури аналізу, проектування і розробки алгоритмів за різними системами команд і виконавців.</p>

	ФК19. Знання та розуміння загальних принципів функціонування та архітектури комп'ютерних систем та основ операційних систем, володіння системним та прикладним програмним забезпеченням.																																				
Програмні результати навчання	<p>ПРН1. Вміння застосовувати основи педагогічної майстерності в умовах сучасних глобалізаційних процесів.</p> <p>ПРН3. Вміння будувати та використовувати інформаційні моделі процесів і явищ, застосовувати їх для досліджень з використанням найновіших засобів програмування.</p> <p>ПРН8. Демонструвати знання та розуміння методів інформатики та програмування і забезпечувати їх використання в навчальному процесі.</p> <p>ПРН10. Отримання знань для підготовки та проведення лекційних, лабораторних та навчальних занять; організації та проведення наукової та самостійної роботи.</p> <p>ПРН12. Вміння проводити заняття з курсів як математичного так і програмного спрямування з учнями.</p> <p>ПРН15. Вміння розробляти, вдосконалювати та впроваджувати методики навчання інформаційним технологіям на основі знань методів, засобів, форм навчання.</p>																																				
Ключові слова	Алгоритм, виконавець алгоритму, команда виконавця, операторна схема, властивості алгоритму, блок-схема, Blockly, графічний алгоритм, Python IDLE, turtle, візуалізація виконання, контрольна точка, метод стеження, трасування, Algovis, модель пересування площиною, обхід перешкоди, пошук шляху, хвильовий алгоритм, алгоритм фарбування, алгоритмічна мова, синтаксис мови, семантика мови, денотативна семантика, семантична модель навчання, модель програмування, метод обернених задач, обернена задача аналізу (обчислення, відновлення умови, вибір відповіді), скінченний автомат, інформаційна модель, операційна (конотативна) семантика.																																				
Формат курсу	<p><u>Очний</u> (стаціонарне навчання): проведення лекцій, лабораторних робіт та консультацій в приміщеннях університету, а в умовах форсмажорних обставин – онлайнний на платформі Microsoft Teams.</p> <p><u>Заочний</u> (заочне навчання): проведення настановчих лекцій, лабораторних робіт та консультацій в приміщеннях університету, а в умовах форсмажорних обставин – онлайнний на платформі Microsoft Teams; виконання завдань, лабораторних робіт, консультації, контроль навчання – онлайнний на платформі Microsoft Teams.</p>																																				
Теми	<table border="1"> <thead> <tr> <th>Тиж-день</th> <th>Тема, план, короткі тези</th> <th>Форма заняття</th> <th>Тривалість год очне (заочне)</th> <th>Термін виконання</th> </tr> </thead> <tbody> <tr> <td rowspan="2">1</td> <td>Алгоритм як метод розв'язування задач. Дії, операції. Характеристики алгоритмів. Виконавець алгоритму. Система команд виконавця. Способи опису алгоритмів. Операторні схеми.</td> <td>Лекція</td> <td>2 (0,5)</td> <td></td> </tr> <tr> <td>Опрацювання матеріалів і виконання завдань за змістом теми 1.</td> <td>Лабораторна робота</td> <td>2 (0,5)</td> <td>Наступне лабораторне заняття</td> </tr> <tr> <td rowspan="2">2</td> <td>Інформатика для початківців. Моделі вивчення алгоритмів. Команди, виконавці, система команд виконавця. Формальний виконавець алгоритму.</td> <td>Лекція</td> <td>2 (0,5)</td> <td></td> </tr> <tr> <td>Опрацювання матеріалів і виконання завдань за змістом теми 2.</td> <td>Лабораторна робота</td> <td>2 (0,5)</td> <td>Наступне лабораторне заняття</td> </tr> <tr> <td rowspan="2">3</td> <td>Композиція алгоритмів. Візуальне програмування. Середовище Blockly. Компонування алгоритму в Blockly, перевірка, збереження. Правила роботи в Blockly.</td> <td>Лекція</td> <td>2 (0,5)</td> <td></td> </tr> <tr> <td>Опрацювання матеріалів і виконання завдань за змістом теми 3.</td> <td>Лабораторна робота</td> <td>2 (0,5)</td> <td>Наступне лабораторне заняття</td> </tr> </tbody> </table>					Тиж-день	Тема, план, короткі тези	Форма заняття	Тривалість год очне (заочне)	Термін виконання	1	Алгоритм як метод розв'язування задач. Дії, операції. Характеристики алгоритмів. Виконавець алгоритму. Система команд виконавця. Способи опису алгоритмів. Операторні схеми.	Лекція	2 (0,5)		Опрацювання матеріалів і виконання завдань за змістом теми 1.	Лабораторна робота	2 (0,5)	Наступне лабораторне заняття	2	Інформатика для початківців. Моделі вивчення алгоритмів. Команди, виконавці, система команд виконавця. Формальний виконавець алгоритму.	Лекція	2 (0,5)		Опрацювання матеріалів і виконання завдань за змістом теми 2.	Лабораторна робота	2 (0,5)	Наступне лабораторне заняття	3	Композиція алгоритмів. Візуальне програмування. Середовище Blockly. Компонування алгоритму в Blockly, перевірка, збереження. Правила роботи в Blockly.	Лекція	2 (0,5)		Опрацювання матеріалів і виконання завдань за змістом теми 3.	Лабораторна робота	2 (0,5)	Наступне лабораторне заняття
Тиж-день	Тема, план, короткі тези	Форма заняття	Тривалість год очне (заочне)	Термін виконання																																	
1	Алгоритм як метод розв'язування задач. Дії, операції. Характеристики алгоритмів. Виконавець алгоритму. Система команд виконавця. Способи опису алгоритмів. Операторні схеми.	Лекція	2 (0,5)																																		
	Опрацювання матеріалів і виконання завдань за змістом теми 1.	Лабораторна робота	2 (0,5)	Наступне лабораторне заняття																																	
2	Інформатика для початківців. Моделі вивчення алгоритмів. Команди, виконавці, система команд виконавця. Формальний виконавець алгоритму.	Лекція	2 (0,5)																																		
	Опрацювання матеріалів і виконання завдань за змістом теми 2.	Лабораторна робота	2 (0,5)	Наступне лабораторне заняття																																	
3	Композиція алгоритмів. Візуальне програмування. Середовище Blockly. Компонування алгоритму в Blockly, перевірка, збереження. Правила роботи в Blockly.	Лекція	2 (0,5)																																		
	Опрацювання матеріалів і виконання завдань за змістом теми 3.	Лабораторна робота	2 (0,5)	Наступне лабораторне заняття																																	

4	Графічні алгоритми і виконавці. Знайомство з графічним виконавцем turtle. Система координат черепахи. Програмовані алгоритми. Огляд команд черепахи.	Лекція	2 (1)	
	Опрацювання матеріалів і виконання завдань за змістом теми 4.	Лабораторна робота	2 (1)	Наступне лабораторне заняття
5	Візуалізація виконання алгоритма. Проста візуалізація в Python методом контрольних точок. Візуалізація контрольними точками як функціями. Контрольна точка з паузою. Метод стеження. Метод логічної прокрутки – трасування. Візуалізація алгоритма в середовищі C++. Графічна візуалізація алгоритма.	Лекція	2 (1)	
	Опрацювання матеріалів і виконання завдань за змістом теми 5.	Лабораторна робота	2 (1)	Наступне лабораторне заняття
6	Алгоритми задач пересування на площині. Вступ до роботехніки. Пересування в умовах невизначеності. Будова площини для задач пересування. Площина з ізольованими опуклими перешкодами. Правосторонній обхід. Система команд виконавця. Лівосторонній обхід. Опуклі перешкоди дотичні до границь. Неопуклі перешкоди.	Лекція	2 (1)	
	Опрацювання матеріалів і виконання завдань за змістом теми 6.	Лабораторна робота	2 (1)	Наступне лабораторне заняття
7	Алгоритми задач пошуку шляху. Модель середовища пересування і пошуку шляхів. Підготовка даних для програмування і виконання алгоритмів. Хвильовий алгоритм. Алгоритм зафарбування замкненої ділянки.	Лекція	2 (1)	
	Опрацювання матеріалів і виконання завдань за змістом теми 7.	Лабораторна робота	2 (1)	Наступне лабораторне заняття
8	Алгоритмічні мови. Високорівневі мови програмування. Режими виконання алгоритмічних мов. Будова синтаксису алгоритмічних мов. Регулярні вирази. Нотація Бекуса-Наура. Класичний спосіб зображення синтаксису. Розширена форма Бекуса-Наура і розбір граматики. Планування дидактичного матеріалу для навчання. Семантика мови. Загальні означення. Приклад часткового визначення денотативної семантики.	Лекція	2 (1)	
	Опрацювання матеріалів і виконання завдань за змістом теми 8.	Лабораторна робота	2 (1)	Наступне лабораторне заняття
9	Семантичні моделі вивчення програмування. Постановка задачі. Денотативна семантика. Синтаксис підмножини мови. Денотативна семантика для навчання C++ на рівні моделей. Моделі програмування на основі операцій. Цілочисельна арифметика. Моделі програмування на основі операторів і керуючих структур.	Лекція	2 (1)	
	Опрацювання матеріалів і виконання завдань за змістом теми 9.	Лабораторна робота	2 (1)	Наступне лабораторне заняття
10	Обернені задачі будови і вивчення алгоритмів. Постановка оберненої задачі навчання. Початкове навчання. Обернені задачі. Приклади задач аналізу. Приклади задач синтезу. Фахове навчання. Обернені задачі на обчислення. Обернені задачі на відновлення умови. Комплексні обернені задачі. Обернені задачі на обчислення з вибором відповіді.	Лекція	2 (0,5)	
	Опрацювання матеріалів і виконання завдань за змістом теми 10.	Лабораторна робота	2 (0,5)	Наступне лабораторне заняття
11-12	Алгоритми і скінченні автомати. Про теорію автоматів. Акцептори і розпізнавачі. Перетворювачі (трансдуктори). Автомат Мура. Автомат Мілі. Приклади задач розпізнавання. Порівняння автоматів Мура і Мілі. Суміщення автоматів Мура і Мілі. Детермінований скінченний автомат. Табличний спосіб визначення автомата Мілі.	Лекція	4 (1)	
	Опрацювання матеріалів і виконання завдань за змістом теми 11-12.	Лабораторна робота	4 (1)	Наступне лабораторне заняття
13	Інформаційні моделі в задачах. Інформаційні моделі і скінченні автомати. Операційна семантика. Обчислювальні задачі. Систематичні вирази. Задачі синтезу (генерації). Задачі графічного аналізу матричних даних.	Лекція	2 (0,5)	

	Опрацювання матеріалів і виконання завдань за змістом теми 13.	Лабораторна робота	2 (0,5)	Наступне лабораторне заняття
14	Методи розробки алгоритмів. Рекурсія. Метод частинних цілей. Динамічне програмування. Програмування з поверненнями назад. Метод сходження. Генетичні алгоритми. Алгоритми на графах. Сортування. Пошук.	Лекція	2 (1)	
	Опрацювання матеріалів і виконання завдань за змістом теми 14.	Лабораторна робота	2 (1)	Наступне лабораторне заняття
15	Олімпіади з інформатики для учнів шкіл. Підготовка учнів до олімпіад. Олімпіадні задачі попередніх років. Умови задач, розбір розв'язків. Критерії оцінювання. Вимоги до будови тестів. Оформлення розв'язків.	Лекція	2 (1)	
	Опрацювання матеріалів і виконання завдань за змістом теми 15.	Лабораторна робота	2 (1)	Наступне лабораторне заняття
16	Організація вивчення інформатики в школі. Особливості вікових груп учнів. Інформатика прикладна і алгоритмічна. Тематичне планування розділів алгоритмізації. Вимоги до результатів навчання за віковими групами.	Лекція	2 (0,5)	
	Опрацювання матеріалів і виконання завдань за змістом теми 16.	Лабораторна робота	2 (0,5)	Початок сесії

Форми контролю

Поточний контроль знань студентів виконується під час проведення лабораторних занять протягом семестру і має на меті перевірку рівня підготовленості студента до виконання завдань за темою, запланованою на окреме кожне заняття. Крім того, поточний контроль виконують для перевірки виконання студентом індивідуальних завдань.

Максимальна кількість балів за поточний контроль – 50.

Екзаменаційний контроль. Іспит в кінці семестру. Екзаменаційний контроль проводиться за розкладом, визначеним навчальним графіком для складання іспитів.

Перелік питань до іспиту викладено вище в розділі "Теми". Задачі до іспиту обирають з числа завдань, виконаних в межах лабораторних робіт. Порядок проведення іспиту, зміст екзаменаційних білетів, критерії оцінювання відповідей будуть оголошені окремо до початку екзаменаційної сесії.

Максимальна кількість балів за екзаменаційний контроль – 50.

Підсумковий контроль включає в себе результати поточного контролю знань студентів, що проводився протягом семестру, та екзаменаційний контроль.

Максимальна кількість балів підсумкового контролю – 100.

Шкала оцінювання підсумкового контролю:

Бали	ECTS	Національна шкала	Коментар
90 - 100	A	5	відмінно
81 - 89	B	4	дуже добре
71 - 80	C	4	добре
61 - 70	D	3	задовільно
51 - 60	E	3	достатньо
26 - 50	FX	2+	незадовільно
0 - 25	F	2-	повторний курс

Пререквізити

Для вивчення курсу студенти потребують базових знань з дисциплін: дискретна математика, програмування, алгоритми і структури даних, програмна інженерія.

<p>Навчальні методи та техніки, які використовують під час викладання курсу</p>	<p><u>Лекції</u> з мультимедійними презентаціями та з демонстрацією прикладів реалізації алгоритмічних моделей за темою лекції; <u>лабораторні</u> заняття у вигляді проєктування алгоритмів і програм на основі визначеної алгоритмічної моделі, виконання практичних завдань та проєктів, мультимедійної <u>презентації</u> студентів; <u>самостійне</u> опрацювання навчальних матеріалів: підручників, конспектів лекцій, електронних ресурсів, готових програм різними алгоритмічними мовами, додаткових матеріалів, розміщених у хмарному сховищі (Microsoft Teams, Google Диск). Вивчення матеріалів <u>веб-сайтів</u> за темами занять. <u>Обговорення</u> теоретичного та практичного матеріалу в онлайн сервісах, <u>формулювання</u> творчих завдань для студентів, виконання яких готує до вивчення нового теоретичного і практичного матеріалу.</p>
<p>Необхідне обладнання</p>	<p>Для проведення лекцій: комп'ютер, проєктор, доступ до мережі інтернет. Для проведення лабораторних та виконання завдань: комп'ютер; ОС Windows/Linux; доступ до інтернету; середовища програмування мовою C++ (Microsoft Visual Studio, Code Blocks тощо), мовою Python (IDLE Python тощо), середовище візуального програмування Blockly, середовища графічної візуалізації алгоритма. Можливі використання інших мов програмування і середовищ розробки та аналізу. Уся література і робочі матеріали, які студенти не зможуть знайти самостійно, буде надана викладачем виключно в освітніх цілях без права її передачі третім особам. Студенти заохочуються до використання також й іншої літератури та джерел, яких немає серед рекомендованих.</p>
<p>Критерії оцінювання (окремо для кожного виду навчальної діяльності)</p>	<p>Оцінювання проводиться за 100-бальною шкалою. 50 балів нараховують за виконання лабораторних завдань та індивідуальних проєктів впродовж семестра, ще 50 балів – за виконання екзаменаційного завдання. Лабораторні завдання чи проєкти індивідуальні. Упродовж семестру студент виконує не менше 10 завдань чи проєктів, які оцінюють різною шкалою залежно від складності (шкала оцінювання і критерії надаються студентам на початку семестра). Оцінка за екзаменаційне завдання може отримати додаткові бали, якщо завдання (проєкти) студента, виконані впродовж семестра, мали закінчений характер і високий рівень фахової реалізації. Відвідання занять є важливою складовою навчання. Очікується, що всі студенти відвідають усі лекції і лабораторні заняття курсу. Активність під час проведення лекцій і лабораторних заохочується балами при оцінюванні відповідного лабораторного завдання чи проєкта. У будь-якому випадку студенти зобов'язані дотримуватися усіх строків визначених для виконання усіх видів робіт, передбачених курсом. Виконані роботи завантажують у відповідне хмарне сховище. Альтернативою відвідування лабораторних занять в університеті може бути дистанційна онлайн робота за розкладом проведення занять. Академічна добросесність: очікується, що роботи студентів будуть їхнім оригінальними дослідженнями чи міркуваннями. Відсутність посилань на використані джерела, фабрикування джерел, списування, втручання в роботу інших студентів, здавання чужих комп'ютерних програм як своїх становлять, але не обмежують, приклади можливої академічної недобросесності. Виявлення ознак академічної недобросесності в письмовій роботі студента є підставою для її незарахування викладачем, незалежно від масштабів плагіату чи обману.</p>
<p>Опитування</p>	<p>Анкету-оцінку з метою оцінювання якості курсу буде надано після завершення курсу.</p>