

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Інститут комп'ютерних технологій, автоматики та метрології (ІКТА)
/назва навчально-наукового інституту/

Кафедра електронних обчислювальних машин (ЕОМ)
/назва /

«ЗАТВЕРДЖУЮ»

Голова науково-методичної комісії
спеціальності 123 “Комп'ютерна інженерія”
/назва /

_____/ Мельник А.О. /
/підпис/ /ініціали та прізвище /

Протокол від « 20 » серпня 2021 р. № 1

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

	СК1.18. Алгоритми та моделі обчислень	
	/код і назва навчальної дисципліни/	
	бакалавр	
	/рівень вищої освіти/	
вид дисципліни	за вибором	
	(обов'язкова / за вибором)	
мова викладання	українська	
освітня програма	ОПП “Комп'ютерна інженерія”	
	/назва/	
галузь знань	12 “Інформаційні технології”	
	/шифр і назва/	
спеціальність	123 “Комп'ютерна інженерія”	
	/шифр і назва /	

Львів – 2021 рік

Робоча програма з навчальної дисципліни Алгоритми та моделі обчислень для
/назва /
здобувачів освіти за освітньою програмою ОПП “Комп’ютерна інженерія”
/ назва освітньої програми /

Розробники:

ст. вик. каф. ЕОМ _____ / Н.Б. Козак /
/посада, науковий ступінь та вчене звання/ _____ /підпис/ _____ /ініціали та прізвище/

Гарант освітньої програми _____ / _____ /
_____ /підпис/ _____ /ініціали та прізвище/

Робоча програма розглянута та схвалена на засіданні кафедри ЕОМ
/назва/

Протокол від «20» серпня 2021 року № 1

Завідувач кафедри ЕОМ _____ / А.О. Мельник /
/назва / _____ /підпис/ _____ /ініціали та прізвище /

1. Структура навчальної дисципліни

Найменування показників	Всього годин	
	Денна форма навчання	Заочна форма навчання
Кількість кредитів/год.	6/180	
Усього годин аудиторної роботи, у т.ч.:	90	
• лекційні заняття, год.	45	
• семінарські заняття, год.	-	
• практичні заняття, год.	15	
• лабораторні заняття, год.	30	
Усього годин самостійної роботи, у т.ч.:	45	
• контрольні роботи, к-сть/год.	-	
• розрахункові (розрахунково-графічні) роботи, к-сть/год.	-	
• індивідуальне науково-дослідне завдання, к-сть/год.	-	
• підготовка до навчальних занять та контрольних заходів, год.	45	
Екзамен	+	
Залік		

Частка аудиторного навчального часу студента у відсотковому вимірі:
денної форми навчання – 50% ; заочної форми навчання – –

2. Мета та завдання навчальної дисципліни

2.1. Мета вивчення навчальної дисципліни та результати навчання

Виробити у студентів чітке та систематизоване уявлення про алгоритми та моделі обчислень.

2.2. Завдання навчальної дисципліни відповідно до освітньої програми

Виробити у студентів здатність застосовувати знання про алгоритми та моделі обчислень.

2.3. Результати навчання відповідно до освітньої програми, методи навчання і викладання, методи оцінювання досягнення результатів навчання

В результаті вивчення курсу студенти повинен оволодіти:

знаннями:

- теорії автоматів,
- теорії формальних мов,
- теорії обчислюваності,
- теорії складності обчислень,
- основних моделей обчислень,
- базових алгоритмів обробки інформації,
- теоретичних основ квантових обчислень;

практичними вміннями:

- відображати та читати алгоритми подані різними методами,
- виконувати аналіз алгоритмів,

- синтезувати алгоритми використовуючи різні алгоритмічні стратегії,
- застосовувати базові алгоритми обробки інформації засобами узагальненого програмування(для мов C++, C# та Java),
- застосовувати функційні моделі обчислень та виконувати програмування комп'ютерних систем застосовуючи парадигму функційного програмування,
- застосовувати паралельні моделі обчислень та виконувати програмування комп'ютерних систем застосовуючи парадигму реактивного програмування,
- застосовувати рівночасні моделі обчислень та виконувати програмування комп'ютерних систем застосовуючи парадигму подійно-орієнтованого програмування.

2.4. Перелік попередніх та супутніх і наступних навчальних дисциплін

№ з/п	Попередні навчальні дисципліни	Супутні і наступні навчальні дисципліни
1.	Дискретна математика	Архітектура комп'ютера
2.	Програмування, частина 1 (Основи алгоритмізації та програмування)	
3.	Програмування, частина 2 (Об'єктно орієнтоване програмування)	

3. Анотація навчальної дисципліни

Курс «Алгоритми та моделі обчислень» розроблений на основі класичних літературних джерел [1, 2, 3, 4, 5, 6] зважаючи на програми аналогічних курсів Массачусетського технологічного інституту [7, 8]. Основу дисципліни становлять наступні розділи.

- Теорія алгоритмів (*в зарубіжній літературі – теорія обчислень, англ. Theory of Computation*) [3]. Тут розглядаються:

- теорія автоматів (*англ. Automata theory*);
- теорія формальних мов (*англ. Formal language theory*);
- теорія обчислюваності (*англ. Computability theory*);
- теорія складності обчислень (*англ. Computational complexity theory*).

- Моделі обчислень (*англ. Models of Computation*) [4, 5].

- Методи розробки алгоритмів(алгоритмічні стратегії)(*англ. Algorithm Design Paradigm, Algorithmic Paradigm, Algorithmic Technique, Algorithmic Strategy*) [6].

Додатково до курсу включений розділ присвячений квантовим обчисленням, які являють собою якісно новий етап розвитку комп'ютерних технологій.

Для базових алгоритмів обробки інформації та бібліотек популярних мов програмування, які їх реалізують, розглядаються зразки коду програм, в яких велику увагу приділено використанню:

- узагальненого програмування;
- метапрограмування;
- регулярних виразів та нотації Бекуса-Наура.

Також окрім імперативного(в основному процедурного та об'єктно-орієнтованого) програмування, в наведених зразках коду показано застосування:

- парадигми функційного програмування;
- парадигми реактивного програмування;
- парадигми подійно-орієнтованого програмування.

4. Опис навчальної дисципліни

4.1. Лекційні заняття

№ з/п	Назви тем	Кількість годин	
		ДФН	ЗФН
1.	<p>Частина 1. Тема №1. Вступ до теорії алгоритмів.</p> <p>1.1. Неформальне тлумачення алгоритму.</p> <p>1.1.1. Історія поняття алгоритму.</p> <p>1.1.2. Визначення алгоритму.</p> <p>1.1.3. Основні властивості алгоритму.</p> <p>1.1.4. Параметри алгоритму.</p> <p>1.1.5. Базові структури алгоритмів(алгоритмічні конструкції).</p> <p>1.2. Формалізація поняття алгоритму.</p> <p>1.2.1. Введення в теорію алгоритмів.</p> <p>1.2.2. Абстрактні моделі алгоритму.</p> <p>1.2.3. Формальні алгоритмічні системи(ФАС).</p> <p>1.2.4. Скінченний автомат.</p> <p>1.2.4.1. Детермінований скінченний автомат(DFA).</p> <p>1.2.4.2. Недетермінований скінченний автомат(NFA).</p> <p>1.2.4.3. Автомат з однобуквеними переходами. Формування автомату з однобуквеними переходами за заданим недетермінованим автоматом.</p> <p>1.2.4.4. Видалення непродуктивних та недосяжних станів скінченного автомату.</p> <p>1.3.4.5. Видалення λ-переходів та ε-переходів у недетермінованому скінченному автоматі.</p> <p>1.3.4.6. Детермінізація квазидетермінованого скінченного автомату.</p> <p>1.3.4.7. Детермінізація недетермінованого скінченного автомату.</p> <p>1.3.4.8. Мінімізація скінченного детермінованого автомату.</p> <p>1.2.5. Перетворювачі(трансдуктори) на основі детермінованого скінченного автомату.</p> <p>1.2.5.1. Автомат Мура(Moore machine).</p> <p>1.2.5.2. Автомат Мілі(Mealy machine).</p> <p>1.2.6. Магазинний автомат(PDA).</p> <p>1.2.7. Машина Тюрінга та теза Черча.</p> <p>1.2.8. Машина Поста.</p> <p>1.2.9. Нормальні алгоритми Маркова.</p> <p>1.3. Формальні граматики та формальні мови.</p> <p>1.3.1. Формальні граматики, мови та ієрархія Чьомські.</p> <p>1.3.2. Регулярні мови та вирази.</p> <p>1.3.2.1. Властивості граматик регулярних мов. Автоматні граматики. Доповнення автоматної мови.</p> <p>1.3.2.2. Лема про накачку для регулярних мов.</p> <p>1.3.2.3. Знаходження мови для заданої регулярної граматики.</p> <p>1.3.2.4. Регулярні вирази.</p> <p>1.3.2.5. Формування регулярного виразу для заданого недетермінованого скінченного автомату.</p> <p>1.3.2.6. Формування недетермінованого скінченного автомату для заданої регулярної граматики.</p> <p>1.3.2.7. Формування недетермінованого скінченного автомату для заданого регулярного виразу.</p> <p>1.3.3. KB-мови та нотації БНФ.</p> <p>1.3.3.1. Властивості граматик KB-мов.</p> <p>1.3.3.2. Лема про накачку для KB-мов.</p> <p>1.3.3.3. Знаходження мови для заданої KB-граматики.</p> <p>1.3.3.4. Нотації БНФ та РБНФ.</p> <p>1.3.3.5. Нормальна форма Хомського для KB-граматик.</p> <p>1.3.3.6. Алгоритм Кока-Касамі-Янгера для KB-граматик у нормальній формі Хомського.</p> <p>1.3.3.7. Створення МП-автомату, який породжується заданою KB-граматикою.</p>	6	

2.	<p>Частина 1. Тема №2. Методи відображення та синтез алгоритмів.</p> <p>2.1. Методи відображення алгоритмів.</p> <p>2.1.1. Вербальне подання алгоритму.</p> <p>2.1.2. Подання алгоритму псевдокодом з використанням формальних мов.</p> <p>2.1.3. Схематичне подання алгоритму.</p> <p>2.1.3.1. Просте графічне подання алгоритму. Блок-схема. ПГА. UML.</p> <p>2.1.3.2. Структурограма. Діаграма Нассі-Шнайдермана.</p> <p>2.2. Синтез алгоритмів.</p> <p>2.2.1. Покрокове проектування алгоритмів.</p> <p>2.2.2. Підходи при синтезі алгоритмів(алгоритмічні стратегії).</p> <p>2.2.2.1. Повний перебір.</p> <p>2.2.2.2. Бектрекінг (перебір з поверненням).</p> <p>2.2.2.3. Метод “розділяй і пануй”.</p> <p>2.2.2.4. Метод гілок і границь.</p> <p>2.2.2.5. Скупі алгоритми.</p> <p>2.2.2.6. Динамічне програмування.</p>	4	
3.	<p>Частина 1. Тема №3. Основи аналізу алгоритмів.</p> <p>3.1. Оцінка розміру вхідних даних. Складність по пам’яті. Складність по часу виконання алгоритму.</p> <p>3.2. Порівняння найкращих, середніх та найгірших оцінок. O-, o-, Ω-, ω-, Θ- нотації. Стандартні класи ефективності алгоритмів.</p> <p>3.3. Математичний аналіз нерекурсивних алгоритмів.</p> <p>3.4. Математичний аналіз рекурсивних алгоритмів.</p> <p>3.5. P- і NP- класи складності. NP-повнота (теорема Кука).</p> <p>3.6. Стандартні NP-повні проблеми.</p> <p>3.7. ZPP-, RP- і BPP-класи складності.</p> <p>3.8. Емпіричний аналіз алгоритмів.</p>	2	

4.	<p>Частина 1. Тема №4. Базові алгоритми обробки інформації.</p> <p>4.1. Алгоритми пошуку.</p> <ul style="list-style-type: none"> 4.1.1. Послідовний пошук. 4.1.2. Бінарний пошук. 4.1.3. Пошук в лінійних списках. 4.1.4. Задача вибору. 4.1.5. Дерева бінарного пошуку. 4.1.6. Збалансовані дерева. 4.1.7. Вичерпний пошук. 4.1.8. Порозрядний пошук. 4.1.9. Зовнішній пошук. 4.1.10. Хешування. 4.1.11. Розв'язання колізій при хешуванні відкритою адресацією та методом ланцюжків. <p>4.2. Алгоритми сортування даних.</p> <ul style="list-style-type: none"> 4.2.1. Сортування вибором. 4.2.2. Сортування вставками. 4.2.3. Сортування обміном. 4.2.4. Сортування злиттям. 4.2.5. Сортування Шелла. 4.2.6. Швидке сортування. 4.2.7. Пірамідальне сортування. 4.2.8. Порозрядне та бітове сортування. 4.2.9. Мережі сортування. Зовнішнє сортування. <p>4.3. Алгоритми порівняння зі взірцем.</p> <ul style="list-style-type: none"> 4.3.1. Алгоритм Рабіна-Карпа. 4.3.2. Пошук підрядків за допомогою скінчених автоматів. 4.3.3. Алгоритм Кнута-Морріса-Пратта. 4.3.4. Алгоритм Бойера-Мура. 4.3.5. Наближене порівняння рядків. <p>4.4. Чисельні алгоритми.</p> <ul style="list-style-type: none"> 4.4.1. Матриці та дії з ними. Множення матриць по Винограду та по Штрассену. 4.4.2. Робота з довгими числами. 4.4.3. Алгебраїчні системи. 4.4.4. Розв'язання систем лінійних рівнянь. 4.4.5. Розв'язання нелінійних рівнянь. 4.4.6. Многочлени та швидке перетворення Фур'є. 4.4.7. Алгоритми апроксимації і інтерполяція чисельних функцій. <p>4.5. Графи та мережеві алгоритми.</p> <ul style="list-style-type: none"> 4.5.1. Пошук у графі. 4.5.2. Породження всіх каркасів графа. 4.5.3. Каркас мінімальної ваги. Метод Дж. Крускала. Метод Р. Пріма. 4.5.4. Досяжність. Визначення зв'язності. Двозв'язність. 4.5.5. Ейлерові цикли. 4.5.6. Гамільтонові цикли. 4.5.7. Фундаментальна множина циклів. 4.5.8. Алгоритм Дейкстри. 4.5.9. Алгоритм Флойда. 4.5.10. Метод генерації всіх максимальних незалежних множин графа. 4.5.11. Задача про найменше покриття. 4.5.12. Розфарбування графа. 4.5.13. Пошук мінімального розфарбування вершин графа. 4.5.14. Використання задачі про найменше покриття при розфарбуванні вершин графа. 4.5.15. Потоки в мережах. 4.5.16. Метод побудови максимального потоку в мережі. 4.5.17. Методи наближеного рішення задачі комівояжера(метод локальної оптимізації, алгоритм Ейлера, алгоритм Крістофідеса). 4.5.18. Аналіз алгоритмів на графах. <p>4.6. Паралельні та розподілені алгоритми.</p> <ul style="list-style-type: none"> 4.6.1. Модель паралельного виконання програми зі спільною пам'яттю і модель передачі повідомлень. 4.6.2. Організація паралельних обчислень відповідно до принципу консенсусу і на основі вибору. 4.6.3. Методи визначення завершення паралельних обчислень. 4.6.4. Паралельний пошук, паралельне сортування, паралельні чисельні алгоритми, паралельні алгоритми на графах. 	12
----	--	----

5.	<p>Частина 1. Тема №5. Бібліотеки основних алгоритмів обробки інформації для популярних мов програмування.</p> <p>5.1. Застосування базових алгоритмів при узагальненому програмуванні на C++ засобами STL(Standart Template Library).</p> <p>5.2. Застосування базових алгоритмів при узагальненому програмуванні на C++ засобами Boost.</p> <p>5.3. Застосування базових алгоритмів при узагальненому програмуванні на Java засобами JCL(Java Class Library).</p> <p>5.4. Застосування базових алгоритмів при узагальненому програмуванні на C# засобами FCL(Framework Class Library).</p> <p>5.5. Застосування алгоритмів лінійної алгебри при програмуванні на C++ за допомогою BLAS.</p> <p>5.6. Застосування алгоритмів обробки сигналів при програмуванні на C++ та Python засобами OpenCV(Open Source Computer Vision Library).</p>	6	
6.	<p>Частина 2. Тема 6. Моделі обчислень.</p> <p>6.1. Послідовні формальні алгоритмічні системи еквівалентні машині Тюрінга.</p> <p>6.1.1. Регістрова машина.</p> <p>6.1.2. РАМ-машина.</p> <p>6.2. Функційні моделі обчислень та парадигма функційного програмування.</p> <p>6.2.1. Функційні моделі обчислень.</p> <p>6.2.1.1. Лямбда числення.</p> <p>6.2.1.2. Типізоване лямбда числення.</p> <p>6.2.1.3. Рекурсивні функції.</p> <p>6.2.1.4. Комбінаційна логіка(як функційна модель обчислень).</p> <p>6.2.2. Парадигма функційного програмування.</p> <p>6.3. Паралельні моделі обчислень та парадигма реактивного програмування.</p> <p>Паралельне програмування.</p> <p>6.3.1. Паралельні моделі обчислень.</p> <p>6.3.1.1. ПРАМ-машина. ПРАМ-машина на основі ПВДН.</p> <p>6.3.1.2. Мережа процесів Кана.</p> <p>6.3.1.3. Мережа Петрі.</p> <p>6.3.1.4. Синхронний потік даних.</p> <p>6.3.2. Парадигма реактивного програмування.</p> <p>6.3.3. Паралельне програмування.</p> <p>6.4. Шаблони проектування програмного забезпечення.</p> <p>6.4.1. Використання шаблонів при проектуванні програмного забезпечення.</p> <p>6.4.2. GoF-шаблони.</p> <p>6.4.2.1. Твірні шаблони(Creational pattern).</p> <p>6.4.2.2. Структурні шаблони(Structural pattern).</p> <p>6.4.2.3. Поведінкові шаблони(Behavioral pattern).</p> <p>6.4.3. GRASP-шаблони.</p> <p>6.4.4. Шаблони рівночасних обчислень(Concurrency pattern).</p> <p>6.4.5. Шаблони архітектури програмного забезпечення(Architectural pattern).</p> <p>6.4.6. Шаблони ядра Linux.</p>	10	
7.	<p>Частина 3. Тема №7. Квантові обчислення.</p> <p>7.1. Основні поняття і принципи квантових обчислень. Квантова машина Тюрінга і VQP-клас складності.</p> <p>7.2. Кубіти, квантові вентиля та квантові регістри.</p> <p>7.2.1. Вимірювання значень кубітів.</p> <p>7.2.2. Система кубітів та квантова запутаність.</p> <p>7.2.3. Вентиль тотожного перетворення.</p> <p>7.2.4. Вентиль заперечення.</p> <p>7.2.5. Вентиль фазового зміщення.</p> <p>7.2.6. Вентиль перетворення Адамара.</p> <p>7.2.7. Прямий керуючий вентиль.</p> <p>7.2.8. Вентиль контрольованого заперечення.</p> <p>7.2.9. Вентиль Тоффолі.</p> <p>7.2.10. Вентиль Фредкіна.</p> <p>7.3. Функціонування квантової системи.</p> <p>7.3.1. Задача Дойча.</p> <p>7.3.2. Задача Дойча-Джозі.</p> <p>7.3.3. Задача Бернштайна-Вазірані.</p> <p>7.3.4. Задача Саймона.</p> <p>7.4. Важливі алгоритми квантових обчислень.</p> <p>7.4.1. Алгоритм Шора.</p> <p>7.4.2. Алгоритм Гровера.</p>	5	
Усього годин		45	

4.2. Практичні та лабораторні заняття

4.2.1. Практичні заняття

№ з/п	Назви тем	Кількість годин	
		ДФН	ЗФН
1	Практичне заняття №1. Вступне заняття. Огляд матеріалу практичних занять. Поняття алгоритму.	2	
2.	Практичне заняття №2. Алгоритм; властивості, параметри та характеристики складності алгоритму. <u>Задання:</u> Порівняти складність арифметичних операцій в римській та десятковій системах числення. <i>(виконується як домашня робота)</i>	2	
3.	Практичне заняття №3. Вплив правила безпосереднього перероблення на характеристики складності алгоритму. <u>Задання:</u> порівняти часову складність трьох алгоритмів знаходження НСД. <i>(виконується як домашня робота)</i>	2	
4.	Практичне заняття №4. Параметри алгоритму. Правило безпосереднього перероблення. Асимптотичні характеристики складності алгоритму. Алгоритми з поліноміальною та експоненціальною складністю. <u>Задання:</u> Визначити часову складність заданого алгоритму в “найгіршому випадку”. <i>(виконується як контрольна робота в кінці пари)</i>	2	
5.	Практичне заняття №5. Використання потокового графу алгоритму при проектуванні паралельних обчислень. <u>Задання:</u> розпаралелити алгоритм шляхом формування потокового графу алгоритму. <i>(виконується як контрольна робота в кінці пари)</i>	2	
6.	Практичне заняття №6. Формальні алгоритмічні системи (ФАС). Машина Тюрінга (МТ). <u>Задання:</u> Побудувати алгоритм для МТ(сформувати “слід” МТ). Підрахувати часову, програмну та місткісну складність. <i>(виконується як контрольна робота в кінці пари)</i>	2	
7.	Практичне заняття №7. Побудова алгоритмів ефективних за часовою складністю. <u>Задання:</u> застосувати метод «гілок і границь» при побудові алгоритму для вирішення задачі квадратичного призначення. <i>(виконується як контрольна робота в кінці пари)</i>	2	
8.	Захист домашніх завдань.	1	
Усього годин		15	

4.2.2. Лабораторні заняття

№ з/п	Назви тем	Кількість годин	
		ДФН	ЗФН
1	Вступне заняття. Інструктаж. Видача завдань.	4	
2.	Виконання лабораторної №1. Алгоритм; властивості, параметри та характеристики складності алгоритму.	4	
3.	Виконання лабораторної №2. Асимптотичні характеристики складності алгоритму; алгоритми з поліноміальною та	4	

	експоненціальною складністю.		
4.	Виконання лабораторної №3. Використання потокового графу алгоритму при проектуванні паралельних обчислень.	4	
5.	Виконання лабораторної №4. Побудова алгоритмів ефективних за часовою складністю; задача квадратичного призначення.	4	
6.	Виконання лабораторної №5. Функційне програмування.	4	
7.	Виконання лабораторної №6. Реактивне програмування.	4	
8.	Заключне заняття. Кінцевий термін захисту лабораторних робіт.	2	
Усього годин		30	

4.3. Самостійна робота

№ з/п	Найменування робіт	Кількість годин	
		ДФН	ЗФН
1.	Підготовка до лабораторних занять.	10	
2.	Підготовка до навчальних занять та контрольних заходів	20	
3.	Самостійне опрацювання теоретичного матеріалу	15	
Усього годин		45	

5. Опис методів оцінювання рівня досягнення результатів навчання

Оцінювання знань студентів з дисципліни “Алгоритми та моделі обчислень” проводиться відповідно до робочого навчального плану у вигляді **семестрового контролю**, який проводиться в кінці семестру і включає в себе результати **поточного контролю** знань студентів, який оцінюється за виконання лабораторних робіт, та **контрольного заходу** – відповідь на відповідний білет на іспиті. Контрольний захід є обов’язковим видом контролю і проводиться в письмово-усній формі в кінці семестру.

Поточний контроль на лекційних заняттях проводиться з метою виявлення готовності студента до занять у таких формах:

- вибіркове усне опитування перед початком занять;
- оцінка активності студента у процесі занять, внесених пропозицій, оригінальних рішень, уточнень і визначень, доповнень попередніх відповідей і т. ін.

Контрольні запитання поділяються на:

- а) тестові завдання – вибрати вірні відповіді;
- б) проблемні – створення ситуацій проблемного характеру;
- в) питання-репліки – виявити причинно-наслідкові зв’язки;
- г) ситуаційні завдання – визначити відповідь згідно певної ситуації;
- д) питання репродуктивного характеру – визначення практичного значення.

6. Критерії оцінювання результатів навчання здобувачів освіти

Максимальна оцінка в балах				
Поточний контроль (ПК)		Екзаменаційний контроль		Разом за дисципліну
Лаб. роботи	Разом за ПК	письмова компонента	усна компонента	
30	30	60	10	100

Порядок та критерії виставляння балів та оцінок:

1. Розподіл балів при умові виконання навчального плану, виконання усіх контрольних робіт і календарного плану виконання лабораторних робіт, інакше за результатами проведення семестрового контролю студент вважається не атестованим.
2. Максимальна кількість балів для оцінки поточного контролю (ПК) знань за семестр – 40 балів.
3. Екзаменаційний контроль проводиться в письмово-усній формі.
4. Максимальна кількість балів для оцінки екзаменаційного контролю – 70 балів.
5. Іспит перед комісією студент складає також в письмово-усній формі з фіксацією запитань та оцінок відповідей на екзаменаційному листі.
6. До іспиту студенти допускаються при умові виконання навчального плану (в тому числі усіх лабораторних робіт).

7. Рекомендована література

1. Cormen, Thomas H.; Leiserson, Charles E.; Rivest, Ronald L.; Stein, Clifford. Introduction to Algorithms. — 3rd. — MIT Press, 2009. — ISBN 0-262-03384-4.
2. Donald E. Knuth. The Art of Computer Programming, Volumes 1-4A Boxed Set. Third Edition (Reading, Massachusetts: Addison-Wesley, 2011), 3168pp. ISBN 978-0-321-75104-1, 0-321-75104-3.
3. Michael Sipser (2013). Introduction to the Theory of Computation. 3rd. Cengage Learning. ISBN 978-1-133-18779-0
4. Savage, John E. (1998). Models Of Computation: Exploring the Power of Computing. ISBN 978-0-201-89539-1
5. Fernandez, Maribel (2009). Models of Computation: An Introduction to Computability Theory. Undergraduate Topics in Computer Science. Springer. ISBN 978-1-84882-433-1.
6. Anany Levitin (2012). Introduction to the design & analysis of algorithms. 3rd. ISBN-13: 978-0-13-231681-1
7. <https://ocw.mit.edu/courses/mathematics/18-404j-theory-of-computation-fall-2006/>
8. <https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-045j-automata-computability-and-complexity-spring-2011/>

8. Політика щодо академічної доброчесності

Політика щодо академічної доброчесності учасників освітнього процесу формується на основі дотримання принципів академічної доброчесності з урахуванням норм «Положення про академічну доброчесність у Національному університеті «Львівська політехніка» (затверджене вченою радою університету від 20.06.2017 р., протокол № 35).

9. УНІФІКОВАНИЙ ДОДАТОК

Національний університет «Львівська політехніка» забезпечує реалізацію права осіб з особливими освітніми потребами на здобуття вищої освіти. Інклюзивні освітні послуги надає Служба доступності до можливостей навчання «Без обмежень», метою діяльності якої є забезпечення постійного індивідуального супроводу навчального процесу здобувачів освіти з інвалідністю та хронічними захворюваннями. Важливим інструментом імплементації інклюзивної освітньої політики в Університеті є Програма підвищення кваліфікації науково-педагогічних працівників та навчально-допоміжного персоналу у сфері соціальної інклюзії та інклюзивної освіти. Звертатися за адресою:

вул. Карпінського, 2/4, І-й н.к., кімн. 112

E-mail: nolimits@lpnu.ua

Websites: <https://lpnu.ua/nolimits>, <https://lpnu.ua/integration>

10. Зміни та доповнення до робочої програми навчальної дисципліни

№ з/п	Зміст внесених змін (доповнень)	Дата і № протоколу засідання кафедри	Примітки