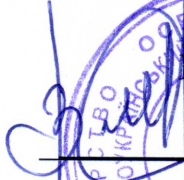


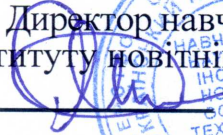
МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

ЗАТВЕРДЖУЮ
Декан факультету комп'ютерних
інформаційних технологій
 Ігор ЯКИМЕНКО

“ 29 ”  2025 р.

ЗАТВЕРДЖУЮ
Проректор з науково-
педагогічної роботи
 Віктор ОСТРОВЕРХОВ

“ 29 ”  2025 р.

ЗАТВЕРДЖУЮ
Директор навчально-наукового
інституту новітніх освітніх технологій
 Святослав ПИТЕЛЬ

“ 29 ”  2025 р.

РОБОЧА ПРОГРАМА

з дисципліни «Проектування комп'ютерних систем на основі графічних
процесорів»

ступінь вищої освіти – бакалавр

галузь знань – 12 “Інформаційні технології”

спеціальність – 123 “Комп'ютерна інженерія”

освітньо-професійна програма – “Комп'ютерна інженерія”

Кафедра комп'ютерної інженерії

Форма навчання	Курс	Семестр	Лекції (год.)	Лабораторні (год.)	ІРС (год.)	Тренінг (год.)	Самост. робота студ. (год.)	Разом (год.)	Екз. (сем.)
Денна	4	8	40	40	5	10	55	150	8
Заочна	4	8	8	4	0	0	138	150	8

Тернопіль – ЗУНУ
2025

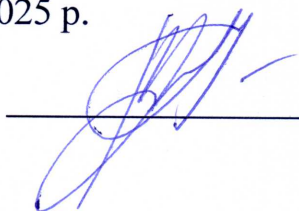
Робоча програма складена на основі освітньо – професійної програми підготовки бакалавра галузі знань 12 “Інформаційні технології” спеціальності 123 “Комп’ютерна інженерія”, затвердженої Вченою радою ЗУНУ (протокол № 9 від 15.06.2022 р.)

Робочу програму склав к.т.н., доцент кафедри КІ

Олег ПІЦУН

Робоча програма затверджена на засіданні кафедри комп’ютерної інженерії, протокол №1 від 26 серпня 2025 р.

Завідувач кафедри



Леся ДУБЧАК

Гарант ОП



Леся ДУБЧАК

СТРУКТУРА РОБОЧОЇ ПРОГРАМИ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
"ПРОЕКТУВАННЯ КОМП'ЮТЕРНИХ СИСТЕМ НА ОСНОВІ ГРАФІЧНИХ
ПРОЦЕСОРІВ"

1. Опис дисципліни «Проектування комп'ютерних систем на основі графічних процесорів»

Дисципліна «Проектування комп'ютерних систем на основі графічних процесорів»	Галузь знань, спеціальність, ступінь вищої освіти	Характеристика навчальної дисципліни
Кількість кредитів – 5	Галузь знань – 12 „Інформаційні технології”	Статус дисципліни – вибіркова Мова навчання – українська
Кількість залікових модулів: 4	Спеціальність – 123 „Комп'ютерна інженерія”	Рік підготовки: <i>Денна</i> - 4 <i>Заочна</i> – 4 Семестр: <i>Денна</i> – 8 <i>Заочна</i> – 8
Кількість змістових модулів – 4	Ступінь вищої освіти - бакалавр	Лекції: <i>Денна</i> - 40 год., <i>Заочна</i> – 8 год. Лабораторні заняття: <i>Денна</i> - 40 год. <i>Заочна</i> – 4 год.
Загальна кількість годин – <i>Денна</i> – 150 год., <i>Заочна</i> - 150 год.		Самостійна робота: <i>Денна</i> – 55 год. <i>Заочна</i> – 138 год. Тренінг– 10 год. Індивідуальна робота: <i>Денна</i> - 5 год.
Тижневих годин: <i>Денна</i> : 1 семестр – 12 год., з них аудиторних – 3 год.		Вид підсумкового контролю <i>Денна</i> : 8 семестр – екзамен <i>Заочна</i> : 8 семестр – екзамен

2. Мета й завдання дисципліни

2.1. Мета вивчення дисципліни

Метою курсу «Проектування комп'ютерних систем на основі графічних процесорів» є отримання знань та навиків розробки програмних систем, які призначені для обробки з допомогою графічних процесорів.

В процесі роботи студенти отримають знання щодо принципів обробки даних з допомогою центрального та графічного процесорів для розуміння принципу роботи сучасних систем. Додатково студенти отримують знання щодо підходів до розробки розпаралелених алгоритмів та отримають практичні навички для програмної реалізації.

2.2 Завдання вивчення дисципліни

Завданням дисципліни визначається тим, щоб дати студентам теоретичну та практичну підготовку із проектування, розробки та програмної реалізації програмних систем з допомогою графічних процесорів.

2.3. Найменування та опис компетентностей, формування котрих забезпечує вивчення дисципліни:

К29. Здатність розуміти та використовувати сучасні парадигми та підходи до розпаралелення алгоритмів для їх подальшого виконання на графічних процесорах.

К30. Здатність застосовувати знання про сучасні методи і мови програмування для розроблення алгоритмічного та програмного забезпечення на основі графічних процесорів.

2.4. Передумови для вивчення дисципліни

Зазначена дисципліна включена до циклу дисциплін професійної підготовки за переліком програми. У структурно-логічній схемі навчання зазначена дисципліна розміщена на III-му курсі. Вивчення курсу Проектування комп'ютерних систем на основі графічних процесорів " передбачає наявність систематичних та ґрунтовних знань із курсів («Програмування», «Мережеве програмування», «Комп'ютерні системи», «Комп'ютерні мережі», «Паралельні та розподілені комп'ютерні системи»), цілеспрямованої роботи над вивченням спеціальної літератури, активної роботи на лекціях та практичних заняттях, самостійної роботи.

2.5. Результати навчання

В результаті вивчення дисципліни студенти повинні:

ПРН24. Вміти розробляти спеціалізоване програмне забезпечення, застосовувати знання технічних характеристик, конструктивних особливостей, призначення і правил експлуатації програмно -технічних засобів комп'ютерних систем та мереж для вирішення технічних задач спеціальності.

ПРН25. Здатність розробляти програмне забезпечення для вбудованих і розподілених застосувань, мобільних і гібридних систем, розраховувати, експлуатувати, типове для спеціальності обладнання.

3. Програма навчальної дисципліни

«Проектування комп'ютерних систем на основі графічних процесорів»

Змістовий модуль 1. Вступ до складності обчислювальних задач і технології CUDA

Тема 1. Вступ. Поняття складності обчислювальних задач

Класифікація обчислювальних задач. Приклади складних обчислювальних задач. Визначення обчислювальної складності задач. Часові та просторові обмеження.

Література: 1, 2.

Тема 2. Підходи до декомпозиції складних задач

Метод декомпозиції задач. Розбиття задач на підзадачі. Принципи модульного підходу. Визначення граничних умов.

Література: 1, 8.

Тема 3. Підходи до виконання складних програм

Паралельні та послідовні програми. Стратегії ефективного виконання програм. Використання багатоядерних систем. Особливості багатопотокових систем.

Література: 1, 9.

Тема 4. Технологія CUDA: Вступ і архітектура

Вступ до CUDA. Архітектура CUDA. Роль графічних процесорів у паралельних обчисленнях. Переваги використання GPU.

Література: 4, 5.

Тема 5. Вступ до моделей паралельних обчислень

Принципи паралельних обчислень. Загальна архітектура паралельних систем. Основні поняття багатопроцесорних систем. Переваги та обмеження паралелізму.

Література: 3, 11.

Змістовий модуль 2. Паралельні обчислення та багатопоточність

Тема 6. Види розпаралелення обчислювальних задач

Поняття паралелізму. Рівні розпаралелення: блоки, потоки. Типи паралельних задач. Масштабування обчислювальних процесів.

Література: 6, 11.

Тема 7. Синхронізація, потоки та процеси

Визначення потоків та процесів. Основні підходи до синхронізації. Засоби для синхронізації процесів. Приклади використання потоків.

Література: 7.

Тема 8. Складність обчислень паралельних задач

Оцінка складності паралельних алгоритмів. Часові та ресурсні обмеження. Співвідношення між кількістю процесів і швидкістю виконання. Блокування та конкуренція потоків.

Література: 8.

Тема 9. Багатопоточність: основи та бібліотеки

Моделі багатопоточності. Огляд бібліотек для багатопоточних систем. Взаємодія між потоками в багатопотокових програмах. Приклади багатопоточних програм.

Література: 12.

Тема 10. Паралелізм на рівні даних

Основи паралельних обчислень з використанням даних. Паралельні структури даних. Приклади паралелізму на рівні даних. Оптимізація розподілу даних.

Література: 12.

Змістовий модуль 3. Основи технології CUDA

Тема 11. CUDA: Ядро та функції

Ядро CUDA. Оголошення функцій у CUDA. Взаємодія хоста і девайса. Стратегії для ефективної роботи з функціями.

Література: 7.

Тема 12. CUDA: Потоки та типи пам'яті

Організація потоків у CUDA. Різні типи пам'яті в CUDA. Приклади роботи з потоками. Оптимізація використання пам'яті в GPU.

Література: 8.

Тема 13. CUDA: Блоки та потоки

Структура блоків і потоків у CUDA. Програмування ідентифікаторів блоків і потоків. Взаємодія між блоками. Паралелізм на рівні блоків.

Література: 8.

Тема 14. CUDA: Атомарні операції та синхронізація

Основи атомарних операцій у CUDA. Важливість синхронізації потоків. Підходи до синхронізації на рівні GPU. Приклади застосування атомарних операцій.

Література: 8, 9.

Тема 15. CUDA: Оптимізація обчислень

Методи оптимізації програм на базі CUDA. Визначення вузьких місць в обчислювальних процесах. Стратегії оптимізації пам'яті. Приклади практичної оптимізації.

Література: 10, 11.

Змістовий модуль 4. Практичне використання технології CUDA

Тема 16. CUDA: Обробка зображень

Методи обробки зображень на GPU. Впровадження фільтрів і алгоритмів обробки. Прискорення обробки зображень за допомогою CUDA. Приклади з реальних задач.

Література: 1, 2, 7.

Тема 17. Алгоритми сортування даних з допомогою CUDA

Типи алгоритмів сортування. Використання CUDA для швидкого сортування. Оптимізація процесу сортування на GPU. Приклади практичних задач.

Література: 7.

Тема 18. CUDA: Робота з нейронними мережами

Нейронні мережі та GPU. Використання CUDA для навчання нейронних мереж. Прискорення обчислень у нейронних мережах за допомогою GPU. Огляд бібліотек для нейронних мереж.

Література: 7.

Тема 19. CUDA: Використання для обробки великих даних

Особливості паралельної обробки великих обсягів даних на GPU. Оптимізація завдань великої вибірки з використанням CUDA. Методи зберігання та доступу до великих даних у CUDA. Приклади обробки великих даних у реальних проектах.

Література: 12.

Тема 20. Практичне застосування CUDA у задачах штучного інтелекту

Впровадження обчислень для AI задач на базі CUDA. Оптимізація нейронних мереж для GPU. Розробка програмного забезпечення для AI з використанням GPU. Використання CUDA у реальних AI задачах.

Література: 11, 12.

4. Структура залікового кредиту дисципліни «Проектування комп'ютерних

систем на основі графічних процесорів»

(денна форма навчання)

Тема	Кількість годин					
	Лекції	Лабораторні заняття	Самостійна робота	Індивідуальна робота	Тренінг	Контрольні заходи
Змістовий модуль 1						
Тема 1. Вступ. Поняття складності обчислювальних задач	2	1	10		2	Опитування
Тема 2. Підходи до декомпозиції складних задач	2	1	10			Опитування
Тема 3. Підходи до виконання складних програм	2	1	10	1		Опитування
Тема 4. Технологія CUDA: Вступ і архітектура	2					Опитування
Тема 5. Вступ до моделей паралельних обчислень	2					Опитування
Змістовий модуль 2						
Тема 6. Види розпаралелення обчислювальних задач	2	1	10			Опитування
Тема 7. Синхронізація, потоки та процеси	2	1	10			Опитування
Тема 8. Складність обчислень паралельних задач	2					Опитування
Тема 9. Багатопоточність: основи та бібліотеки	2					Опитування
Тема 10. Паралелізм на рівні даних	2					Опитування
Змістовий модуль 3						
Тема 11. CUDA: Ядро та функції	2	2	9		2	Опитування
Тема 12. CUDA: Потоки та типи пам'яті	2	1	10	1		Опитування
Тема 13. CUDA: Блоки та потоки	2	1	10			Опитування
Тема 14. CUDA: Атомарні операції та синхронізація	2					Опитування
Тема 15. CUDA: Оптимізація обчислень	2					Опитування

Змістовий модуль 4						
Тема 16. CUDA: Обробка зображень	2	1	8		2	Опитування
Тема 17. Алгоритми сортування даних з допомогою CUDA	2	1	9			Опитування
Тема 18. CUDA: Робота з нейронними мережами	2	1	8			Опитування
Тема 19. CUDA: Використання для обробки великих даних	2					Опитування
Тема 20. Практичне застосування CUDA у задачах штучного інтелекту	2					Опитування
Разом	40	40	55	5	10	

(заочна форма навчання)

	Кількість годин		
	Лекції	Лабораторні заняття	Самостійна робота
Тема 1. Вступ. Поняття складності обчислювальних задач	1		10
Тема 2. Підходи до декомпозиції складних задач			6
Тема 3. Підходи до виконання складних програм			6
Тема 4. Технологія CUDA: Вступ і архітектура	1	1	10
Тема 5. Вступ до моделей паралельних обчислень			6
Тема 6. Види розпаралелення обчислювальних задач	1		10
Тема 7. Синхронізація, потоки та процеси	1		10
Тема 8. Складність обчислень паралельних задач			6
Тема 9. Багатопоточність: основи та бібліотеки			8
Тема 10. Паралелізм на рівні даних			6

Тема 11. CUDA: Ядро та функції	1	1	6
Тема 12. CUDA: Потоки та типи пам'яті	1	1	6
Тема 13. CUDA: Блоки та потоки	1	1	6
Тема 14. CUDA: Атомарні операції та синхронізація	1		6
Тема 15. CUDA: Оптимізація обчислень			6
Тема 16. CUDA: Обробка зображень			6
Тема 17. Алгоритми сортування даних з допомогою CUDA			6
Тема 18. CUDA: Робота з нейронними мережами			6
Тема 19. CUDA: Використання для обробки великих даних			6
Тема 20. Практичне застосування CUDA у задачах штучного інтелекту			6
Разом	8	4	138

5. Тематика лабораторних занять

Лабораторна робота №1.

Тема: CUDA: Інсталювання та основи технології.

Мета: Ознайомитись інструментами для запуску задач під CUDA.

Питання для обговорення:

1. Синтаксис Python для запуску CUDA задач
2. запуск задач у хмарному середовищі
3. запуск задач в локальному середовищі

Література: 1, 2.

Лабораторна робота №2.

Тема: CUDA: Ядра та компілювання програм.

Мета: Навчитись писати та запусками власні програми з допомогою CUDA

Питання для обговорення:

1. Функції для роботи зі стрічками
2. Функції для роботи з масивами
3. Функціональний підхід в програмуванні

Література: 1, 4, 8.

Лабораторна робота №3.

Тема: CUDA: Потоки та типи пам'яті.

Мета: Ознайомитись із роботою з потоками.

Питання для обговорення:

1. Поняття потоків
2. Типи пам'яті
3. Патерни проектування

Література: 4, 8, 16.

Лаборобота №4.

Тема: CUDA: Атомарні операції та синхронізація потоків.

Мета: Набуття практичних навичок забезпечення синхронізації потоків в CUDA.

Питання для обговорення:

1. синхронізація потоків
2. види потоків
3. основні задачі синхронізації.

Література: 4, 8.

Лабораторна робота №5.

Тема: CUDA: Робота із зображеннями.

Мета: Навчитись працювати із зображеннями в CUDA

1. Основні бібліотеки CUDA для опрацювання зображень
2. Забезпечення багатопоточності
3. Функції для роботи із зображеннями

Література: 4, 8.

6. Організація і проведення тренінгу з дисципліни «Проектування комп'ютерних систем на основі графічних процесорів»

№п/п	Вид роботи	Порядок проведення тренінгу
1	Аналіз сучасних підходів розпаралелення задач	- вибір середовища програмування - вибір мови програмування
2	Проектування розпаралеленої програми на основі технології CUDA	- постановка задачі; - проектування процесу реалізації завдання - розподіл на рівні потоків
3	Реалізація програмного застосунку	- реалізація завдання у вибраному середовищі

7. Самостійна робота студентів

(денна форма навчання)

№ п/п	Тематика
1	Класифікація обчислювальних кластерів
2	Грид та хмарні обчислення
3	CUDA: Атомарні операції та синхронізація потоків
4	Робота з нейронними мережами на основі технології CUDA
5	CUDA: Потоки та типи пам'яті
6	CUDA: Робота з блоками та потоками
7	Архітектури високопродуктивних обчислювальних систем
8	Класифікація обчислювальних кластерів

9	CUDA: Атомарні операції та синхронізація потоків
10	CUDA: Ядра та компілювання програм
11	CUDA: Обробка зображень
12	Порівняння CPU та GPU для обчислювальних задач
13	Методи оптимізації використання пам'яті в GPU
14	Моделі багатопоточності в графічних процесорах
15	Оптимізація алгоритмів сортування на GPU
16	Обробка великих даних за допомогою графічних процесорів

(заочна форма навчання)

№ п/п	Тематика
1	Класифікація обчислювальних кластерів
2	Грид та хмарні обчислення
3	CUDA: Атомарні операції та синхронізація потоків
4	Робота з нейронними мережами на основі технології CUDA
5	CUDA: Потоки та типи пам'яті
6	CUDA: Робота з блоками та потоками
7	Архітектури високопродуктивних обчислювальних систем
8	Класифікація обчислювальних кластерів
9	CUDA: Атомарні операції та синхронізація потоків
10	CUDA: Ядра та компілювання програм
11	CUDA: Обробка зображень
12	Порівняння CPU та GPU для обчислювальних задач
13	Методи оптимізації використання пам'яті в GPU
14	Моделі багатопоточності в графічних процесорах
15	Оптимізація алгоритмів сортування на GPU
16	Обробка великих даних за допомогою графічних процесорів

8. Засоби оцінювання та методи демонстрування результатів навчання

У навчальному процесі застосовуються: лекції, в тому числі з використанням мультимедіа проектора та інших ТЗН; практичні заняття; індивідуальні заняття, самостійна робота студента, робота в мережі Інтернет.

У процесі вивчення дисципліни "Проектування комп'ютерних систем на основі графічних процесорів" використовуються наступні засоби оцінювання та методи демонстрування результатів навчання:

- поточні опитування;
- модульне тестування;
- презентації результатів виконання завдань та досліджень;
- оцінювання результатів виконання тренінгів;
- оцінювання результатів самостійної роботи;
- екзамен.

9. Критерії, форми поточного та підсумкового контролю

В процесі вивчення дисципліни "Проектування комп'ютерних систем на основі графічних процесорів" використовуються наступні методи оцінювання навчальної роботи студента:

- поточне тестування та опитування;

- підсумкове тестування по кожному змістовому модулю;
- підсумкова оцінка за виконання завдань на тренінгах;
- підсумковий екзамен.

Підсумковий бал (за 100-бальною шкалою) з дисципліни "Проектування комп'ютерних систем на основі графічних процесорів" визначається як середньозважена величина, в залежності від питомої ваги кожної складової залікового кредиту:

8 семестр

Модуль 1		Модуль 2		Модуль 3	Модуль 4
10 %	10 %	10 %	20 %	10 %	40%
Поточне оцінювання	Модульний контроль 1	Поточне оцінювання	Модульний контроль 1	Тренінги	Екзамен
Середнє арифметичне за 3 лабораторних роботи	Тестові завдання	Середнє арифметичне за 2 лабораторних роботи	Письмова робота: 2 теоретичних питання, 1 задача, тестові завдання	Виконання завдання	2 теоретичних питання по 25 балів = 50 балів, Задача = 50 балів

Оцінка за «Поточне оцінювання» визначається як середнє арифметичне оцінок, отриманих студентом при виконанні лабораторних завдань. Кожне завдання оцінюється за 100 бальною шкалою на основі якості виконання та відповідності поставленим завданням. Бали нараховуються відповідно до таких критеріїв:

- 15 балів – розуміння теоретичних відомостей (розкриття понятійного апарату відповідно до теми лабораторного заняття);
- 25 балів – правильність розв'язування (коректність реалізації, правильний вибір підходу);
- 10 балів – якість виконання (унікальність рішення, стиль програмування, якість програмного коду);
- 10 балів – результативність / тестування (програма видає коректний результат при тестуванні);
- 10 балів – аналіз і висновки (опис отриманих результатів, засоби їх досягнення);
- 30 балів – захист практичної роботи (відповідь на запитання викладача).

Модульний контроль проводиться у формі виконання тестових завдань з однією правильною відповіддю. Тестовий контроль складається з теоретичних питань і одного практичного завдання, що передбачає проведення практичних дій перед вибором варіанту відповіді. Частка виконання практичного завдання становить 40% (40 балів при правильній відповіді) у підсумку. Відповідно частка теоретичної частини складає 60% (60 балів). Максимальний бал за виконання теоретичних

питань модульного контролю залежить від середнього арифметичного правильних відповідей (наприклад, якщо тестів 20 і студент відповів на 15 правильно, то результат рівний $15/20 = 0.75$, тобто 75 % від максимально можливого балу).

Підсумкове значення виконання тестового контролю визначається сумою балів отриманих при виконанні тестових і практичних завдань.

Тренінг оцінюється з повнотою реалізації, якістю програмного коду, тестуванням та аналізом ефективності виконання завдань відповідно до п.7 даної робочої програми. Оцінювання рівня виконання завдань тренінгу проводиться викладачем за 100-бальною шкалою згідно таблиці 1.

Таблиця 1 – Критерії оцінювання тренінгу

Завдання	Макс. бал	Високий рівень	Достатній рівень	Середній рівень	Низький рівень
1. Аналіз сучасних підходів розпаралелення задач	30	22-30 аргументоване, з порівняльним аналізом	15-21 достатньо аргументоване, але без альтернатив	10-14 поверхневий аналіз	0-9 відсутнє або невідповідне
2. Проектування розпаралеленої програми на основі технології CUDA	30	22-30 реалізація у повній мірі	15-21 реалізація з незначними недоліками	10-14 передбачено лише часткові випадки	0-9 відсутнє або невідповідне
3. Реалізація програмного застосунку	40	26-40 комплексне тестування	17-25 тестування, часткове	10-16 часткова відповідність, наявність помилок	0-9 відсутнє або невідповідне
Разом	100	-	-	-	-

Модуль «Самостійна робота» оцінюється викладачем за якістю виконаного індивідуального завдання та звіту відповідно до критеріїв, визначених у розділі 6 «Самостійна робота» цієї робочої програми. Підсумковий бал розраховується як сума оцінок за окремими критеріями у таблиці 2.

Таблиця 2 – Критерії оцінювання самостійної роботи

Завдання	Макс. бал	Високий рівень	Достатній рівень	Середній рівень	Низький рівень
----------	-----------	----------------	------------------	-----------------	----------------

Вибір індивідуального завдання і опис предметної області відповідно до індивідуального завдання	20	22-30 алгоритм реалізовано у повній мірі	15-21 алгоритм реалізований в неповній мірі	10-14 поверховий аналіз	0-9 відсутнє або невідповідне
Вибір технології CUDA, додаткових бібліотек	20	22-30 універсальний підхід	15-21 загальний, не універсальний підхід	10-14 типові рішення	0-9 відсутнє або невідповідне
Розробка ярусно-паралельної форми алгоритму	20	26-40 універсальний підхід	17-25 без врахування виключних ситуацій	10-14 часткова відповідність, наявність помилок	0-9 відсутнє або невідповідне
Програмна реалізація модулю відповідно до індивідуального завдання	40	26-40 комплексне тестування	17-25 тестування, часткове	10-14 часткова відповідність, наявність помилок	0-9 відсутнє або невідповідне
Разом	100	-	-	-	-

Отримана оцінка за національною шкалою ECTS виставляється відповідно до шкали оцінювання рівня знань.

Шкала оцінювання:

За шкалою університету	За національною шкалою	За шкалою ECTS
90-100	Відмінно	A (відмінно)
85-89	Добре	B (дуже добре)
75-84		C (добре)
65-74	Задовільно	D (задовільно)
60-64		E (достатньо)
35-59	Незадовільно	FX (незадовільно, з можливістю повторного складання)
1-34		F (незадовільно, з обов'язковим повторним курсом)

10. Інструменти, обладнання та програмне забезпечення, використання яких передбачає навчальна дисципліна

№	Найменування	Номер теми
1.	Google Colaboratory, Python	11-20

Рекомендовні джерела інформації

1. Березький О.М. Методи, алгоритми та програмні засоби опрацювання біомедичних зображень: монографія / [О.М. Березький, Ю.М. Батько, К.М. Березька, С.О. Вербовий, Т.В. Дацко, Л.О. Дубчак, І.В. Ігнатєв, Г.М. Мельник, В.Д. Николіук, О.Й. Піцун]; під наук. ред. Березький О.М., Тернопіль . THEU «Економічна думка», 2017. - 330 с
2. Berezsky O. GPU – based biomedical image processing / O. Berezsky, O. Pitsun, L. Dubchak, P. Lyaschynsky, P. Lyaschynsky // Proceedings of XIV International Conference Perspective Technologies and methods in mems design (MEMSTECH 2018) 18-22 April, 2018, Lviv-Polyana, Ukraine, pp. 96- 99
3. Ryan N Gutenkunst, dadi.CUDA: Accelerating Population Genetics Inference with Graphics Processing Units, Molecular Biology and Evolution, Volume 38, Issue 5, May 2021, Pages 2177–2178.
4. Duane Storti. CUDA for Engineers: An Introduction to High-Performance Parallel Computing 1st Edition/ Addison-Wesley Professional; 1st edition (November 2, 2015) – 452 p.
5. Xu, Derong, Y. Luo, C. Montag, Y. Hao, and J. Qiang. Full Range Tune Scan Studies Using Graphics Processing Units With CUDA in EIC beam-beam simulations. No. BNL-222292-2021-COPA. Brookhaven National Lab.(BNL), Upton, NY (United States), 2021..
6. Brian Tuomanen Hands-On GPU Programming with Python and CUDA: Explore high-performance parallel computing with CUDA 1st Edition, Kindle Edition – 2018
7. Jaeyeun Han, Bharatkumar Sharma, Learn CUDA Programming - Packt Publishing (September 27, 2019) – 508p.
8. Bhaumik Vaidya. Hands-On GPU-Accelerated Computer Vision with OpenCV and CUDA: Effective techniques for processing complex image data in real time using GPUs Kindle Edition - Packt Publishing; 1st edition (September 26, 2018) – 382 p.
9. Pitsun O. Multi-threaded Parallelization of Automatic Immunohistochemical Image Segmentation. In: Hu, Z., Wang, Y., He, M. (eds) Advances in Intelligent Systems, Computer Science and Digital Economics IV. CSDEIS 2022. Lecture Notes on Data Engineering and Communications Technologies, vol 158. Springer, Cham
10. Aslam, Maria, Omer Riaz, Shahzad Mumtaz, and Ali Daniyal Asif. "Performance comparison of GPU-based jacobi solvers using CUDA provided synchronization methods." IEEE Access 8 (2020): 31792-31812.
11. Träff, Jesper Larsson. "Lectures on Parallel Computing." *arXiv preprint arXiv:2407.18795* (2024).
12. Multithreading: <https://ocw.mit.edu/courses/1-124j-foundations-of-software-engineering-fall-2000/pages/lecture-notes/multithreading/>