

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Львівський національний університет імені Івана Франка
Факультет електроніки та комп'ютерних технологій
Кафедра системного проектування

Затверджено

На засіданні кафедри системного
проектування факультету електроніки
та комп'ютерних технологій
Львівського національного університету
імені Івана Франка
(протокол № ____ від _____ 2023 р.)

Завідувач кафедри:

_____ Роман ШУВАР

Силабус з навчальної дисципліни
“Програмування на GPGPU”,
що викладається в межах ОПШ
“Високопродуктивний комп'ютинг”
першого (бакалаврського) рівня вищої освіти для здобувачів з
спеціальності 121 – Інженерія програмного забезпечення

Львів 2023 р.

Назва дисципліни	Програмування на GPGPU
Адреса викладання дисципліни	Корпус факультету електроніки та комп'ютерних технологій Львівський національний університет імені Івана Франка вул. Драгоманова 50, м. Львів, 79005 вул. Ген. Тарнавського 107, м. Львів, 79011
Факультет та кафедра, за якою закріплена дисципліна	Факультет електроніки та комп'ютерних технологій Кафедра системного проектування
Галузь знань, шифр та назва спеціальності	12 – Інформаційні технології 121 – Інженерія програмного забезпечення
Викладачі дисципліни	Парубочий Віталій Олегович, асистент кафедри системного проектування
Контактна інформація викладачів	vitalius.parubochyi@lnu.edu.ua https://electronics.lnu.edu.ua/en/employee/parubochyj-v
Консультації з питань навчання по дисципліні відбуваються	Консультації проводяться в день проведення лекційних та лабораторних занять (за попередньою домовленістю): ауд. 305, корпус факультету електроніки та комп'ютерних технологій, вул. Драгоманова 50, м. Львів. Також можливі онлайн консультації через платформу BigBlueButton в системі Moodle. Для погодження часу онлайн консультацій слід писати на електронну пошту викладача.
Сторінка курсу	https://moodle.elct.lnu.edu.ua/course/view.php?id=242
Інформація про дисципліну	Дисципліна “Програмування на GPGPU” є нормативною дисципліною зі спеціальності 121 “Інженерія програмного забезпечення” освітньої програми “Високопродуктивний комп'ютинг”, яка викладається в 5-му семестрі в обсязі 3,5 кредитів (за Європейською Кредитно-Трансферною Системою ECTS).
Коротка анотація дисципліни	В курсі “Програмування на GPGPU” розглядаються основні поняття, засоби та особливості програмування з використанням технології обчислень загального призначення на графічних процесорах (GPGPU). Основна увага приділяється засвоєнню знань та отриманню навиків розробки програмних застосунків за допомогою технології CUDA та програмного інтерфейсу CUDA C/C++, що використовують засоби графічного процесора для оптимізації та прискорення обчислень, що вимагають значних ресурсів та часу. Предметом вивчення навчальної дисципліни є галузь обчислень загального призначення на графічних процесорах, її поняття, технології та засоби, а також технологія CUDA як найбільш перспективна та розвинута реалізація технології GPGPU. Для закріплення теоретичних відомостей передбачений лабораторний курс.
Мета та цілі дисципліни	<i>Метою дисципліни “Програмування на GPGPU” є надання студентам знань та розуміння термінів обчислень загального призначення на графічних процесорах (GPGPU) та технології CUDA, а також навичок і вмінь для розробки та оптимізації програмних застосунків, що</i>

	<p>використовують засоби графічного процесора для оптимізації та прискорення обчислень.</p> <p>Цілями дисципліни “Програмування на GPGPU” є забезпечення знайомства студентів зі загальною термінологією та засобами обчислень загального призначення на графічних процесорах (GPGPU) та технології CUDA, формування навичок їхнього практичного використання; отримання студентами знань для аналізу практичних задач з метою їхньої оптимізації засобами технології CUDA; отримання студентами навичок і вмінь застосування програмного інтерфейсу, бібліотек, методів та засобів технології CUDA для реалізації програмних застосунків, що використовують засоби графічного процесора для оптимізації та прискорення обчислень.</p>
<p>Література для вивчення дисципліни</p>	<ol style="list-style-type: none"> 1. M. Harris, "A brief history of GPGPU," UNC Ph.D., 2003 [Online]. Available: https://web.archive.org/web/20220318182947/https://www.cs.unc.edu/xcms/wpfiles/50th-symp/Harris.pdf 2. S. Soller, "GPGPU origins and GPU hardware architecture," High performance computing center Stuttgart, Stuttgart Media University, Stuttgart [Online]. Available: https://jeewhanchoi.github.io/publication/pdf/brief_history.pdf 3. R. Vuduc and J. Choi, "A brief history and introduction to GPGPU," Modern accelerator technologies for geographic information science, pp. 9-23, Aug 2013. doi: 10.1007/978-1-4614-8745-6_2. 4. J. Nickolls and D. Kirk, "Graphics and Computing GPUs," in D. A. Patterson and J. L. Hennessy, Computer Organization and Design RISC-V Edition: The Hardware Software Interface, 2nd ed. Waltham, MA, USA: Morgan Kaufmann, 2020, Appendix B. ISBN: 978-0-12-820331-6. 5. T. M. Aamodt, W. W. L. Fung, and T. G. Rogers. <i>General-Purpose Graphics Processor Architectures</i>. Kentfield, CA: Morgan & Claypool Publishers, 2018. ISBN: 978-1-62705-923-7, doi: 10.2200/S00848ED1V01Y201804CAC044. 6. S. Biswas, "GPGPU Architectures and CUDA C," in An Introductory Course on High-Performance Computing in Engineering, Indian Institute of Technology Kanpur, 2019 [Online]. Available: https://www.iitk.ac.in/hpc4e/workshop_files/GPU-CUDA_v3.pdf 7. P. N. Glaskowsky, "NVIDIA's Fermi: The First Complete GPU Computing Architecture," NVIDIA Corporation [Online], 2009. Available: https://www.nvidia.com/content/PDF/fermi_white_papers/P.Glaskowsky_NVIDIA's_Fermi-The_First_Complete_GPU_Architecture.pdf 8. NVIDIA Corporation. NVIDIA CUDA Compute Unified Device Architecture - Programming Guide Version 1.0. (2007). Accessed: Oct. 07, 2023. [Online]. Available: https://developer.download.nvidia.com/compute/cuda/1.0/NVIDIA_CUDA_Programming_Guide_1.0.pdf 9. J. Sanders and E. Kandrot. <i>CUDA by example: an introduction to general-purpose GPU programming</i>. USA: Addison-Wesley Professional, 2010. ISBN 978-0-13-138768-3. 10. J. Luitjens, "CUDA streams: Best Practices and Common Pitfalls," GPU Technology Conference, San Jose Convention Center, CA, 2014 [Online]. Available: <a 513="" 530="" 894="" 910"="" data-label="Page-Footer" href="https://on- </td> </tr> </table> </div> <div data-bbox="> <p>3</p>

	<p>demand.gputechconf.com/gtc/2014/presentations/S4158-cuda-streams-best-practices-common-pitfalls.pdf</p> <p>11. Steve Rennie, "CUDA C/C++ Streams and Concurrency," Webinar, NVIDIA Corporation [Online], 2014. Available: https://developer.download.nvidia.com/CUDA/training/StreamsAndConcurrencyWebinar.pdf</p> <p>12. NVIDIA Corporation. <i>CUDA Toolkit Documentation</i> [Online]. Available: https://docs.nvidia.com/cuda/index.html</p> <p>13. NVIDIA Corporation. <i>CUDA C++ Programming Guide</i> [Online]. Available: https://docs.nvidia.com/cuda/cuda-c-programming-guide/index.html</p> <p>14. NVIDIA Corporation. <i>CUDA Driver API :: CUDA Toolkit Documentation</i> [Online]. Available: https://docs.nvidia.com/cuda/cuda-driver-api/index.html</p> <p>15. NVIDIA Corporation. <i>CUDA Math API :: CUDA Toolkit Documentation</i> [Online]. Available: https://docs.nvidia.com/cuda/cuda-math-api/index.html</p> <p>16. NVIDIA Corporation. <i>cuRAND :: CUDA Toolkit Documentation</i> [Online]. Available: https://docs.nvidia.com/cuda/curand/index.html</p> <p>17. NVIDIA Corporation. <i>cuBLAS :: CUDA Toolkit Documentation</i> [Online]. Available: https://docs.nvidia.com/cuda/cublas/index.html</p> <p>18. NVIDIA Corporation. <i>NVBLAS :: CUDA Toolkit Documentation</i> [Online]. Available: https://docs.nvidia.com/cuda/nvblas/index.html</p> <p>19. NVIDIA Corporation. <i>Thrust :: CUDA Toolkit Documentation</i> [Online]. Available: https://docs.nvidia.com/cuda/thrust/index.html</p> <p>20. NVIDIA Corporation. <i>Thrust: The C++ Parallel Algorithms Library</i> [Online]. Available: https://nvidia.github.io/thrust/</p> <p>21. NVIDIA Corporation. <i>NVIDIA 2D Image And Signal Performance Primitives (NPP) :: CUDA Toolkit Documentation</i> [Online]. Available: https://docs.nvidia.com/cuda/npp/index.html</p> <p>22. Y. Song, "Image Contrast Adjustment using Nvidia Performance Primitives (NPP)," GPU Technology Conference, San Jose Convention Center, CA, 2014 [Online]. Available: https://on-demand.gputechconf.com/gtc/2014/presentations/HANDS-ON-LAB-S4793-image-processing-using-npp.pdf</p> <p>23. NVIDIA Corporation. <i>cuFFT 12.3 documentation</i> [Online]. Available: https://docs.nvidia.com/cuda/cufft/index.html</p> <p>24. NVIDIA Corporation. <i>NVIDIA cuFFTDx - cuFFTDx 1.1.0 documentation</i> [Online]. Available: https://docs.nvidia.com/cuda/cufftdx/index.html</p> <p>25. I. Buck, "The Evolution of GPUs for General Purpose Computing," GPU Technology Conference, San Jose Convention Center, CA, 20-23 September 2010 [Online]. Available: https://www.nvidia.com/content/gtc-2010/pdfs/2275_gtc2010.pdf</p> <p>26. NVIDIA Corporation. <i>NVIDIA Technical Blog</i> [Online]. Available: https://developer.nvidia.com/blog/</p> <p>27. NVIDIA Corporation, NVIDIA Developer. <i>CUDA Code Samples</i> [Online]. Available: https://developer.nvidia.com/cuda-code-samples</p>
Обсяг курсу	Загальний обсяг: 105 годин. З них 32 години лекцій, 32 години лабораторних робіт та 41 година самостійної роботи.
Очікувані	У результаті вивчення цього курсу студент буде:

<p>результати навчання</p>	<p>знати: основні поняття, визначення і проблеми курсу; вимоги та особливості задач, що можуть бути вирішені за допомогою обчислень загального призначення на графічних процесорах; призначення та особливості застосування основних елементів програмного інтерфейсу CUDA C/C++, стандартних бібліотек CUDA та основних інструментів розробки програмних застосунків за допомогою технології CUDA.</p> <p>вміти: володіти основною термінологією обчислень загального призначення на графічних процесорах; застосовувати програмний інтерфейс CUDA C/C++, бібліотеки CUDA, а також інструменти та засоби розробки технології CUDA для реалізації програмних застосунків, що використовують засоби графічного процесора для оптимізації та прискорення обчислень; виконувати оцінку точності та швидкодії реалізованих програмних застосунків; виконувати оцінку ступеня використання ресурсів графічного процесора.</p> <p>Після вивчення курсу “Програмування на GPGPU” здобувачі набудуть таких Загальних та Фахових компетентностей та Програмних результатів навчання:</p> <p>ЗК01. Здатність до абстрактного мислення, аналізу та синтезу. ЗК02. Здатність застосовувати знання у практичних ситуаціях. ЗК05. Здатність вчитися і оволодівати сучасними знаннями. ЗК06. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.</p> <p>ФК13. Здатність ідентифікувати, класифікувати та формулювати вимоги до програмного забезпечення. ФК14. Здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування. ФК15. Здатність розробляти архітектури, модулі та компоненти програмних систем. ФК17. Здатність дотримуватися специфікацій, стандартів, правил і рекомендацій в професійній галузі при реалізації процесів життєвого циклу. ФК19. Володіння знаннями про інформаційні моделі даних, здатність створювати програмне забезпечення для зберігання, видобування та опрацювання даних. ФК20. Здатність застосовувати фундаментальні і міждисциплінарні знання для успішного розв’язання завдань інженерії програмного забезпечення. ФК22. Здатність накопичувати, обробляти та систематизувати професійні знання щодо створення і супроводження програмного забезпечення та визнання важливості навчання протягом всього життя. ФК23. Здатність реалізовувати фази та ітерації життєвого циклу програмних систем та інформаційних технологій на основі відповідних моделей і підходів розробки програмного забезпечення. ФК24. Здатність здійснювати процес інтеграції системи, застосовувати стандарти і процедури управління змінами для підтримки цілісності, загальної функціональності і надійності програмного забезпечення.</p>
-----------------------------------	---

	<p>ФК25. Здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення.</p> <p>ФК26. Здатність до алгоритмічного та логічного мислення.</p> <p>ПР01. Аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки.</p> <p>ПР03. Знати основні процеси, фази та ітерації життєвого циклу програмного забезпечення.</p> <p>ПР04. Знати і застосовувати професійні стандарти і інші нормативно-правові документи в галузі інженерії програмного забезпечення.</p> <p>ПР05. Знати і застосовувати відповідні математичні поняття, методи доменного, системного і об'єктно-орієнтованого аналізу та математичного моделювання для розробки програмного забезпечення.</p> <p>ПР06. Уміння вибирати та використовувати відповідну задачі методологію створення програмного забезпечення.</p> <p>ПР07. Знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення.</p> <p>ПР09. Знати та вміти використовувати методи та засоби збору, формулювання та аналізу вимог до програмного забезпечення.</p> <p>ПР10. Проводити передпроектне обстеження предметної області, системний аналіз об'єкта проектування.</p> <p>ПР11. Вибирати вихідні дані для проектування, керуючись формальними методами опису вимог та моделювання.</p> <p>ПР12. Застосовувати на практиці ефективні підходи щодо проектування програмного забезпечення.</p> <p>ПР13. Знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань.</p> <p>ПР14. Застосовувати на практиці інструментальні програмні засоби доменного аналізу, проектування, тестування, візуалізації, вимірювань та документування програмного забезпечення.</p> <p>ПР15. Мотивовано обирати мови програмування та технології розробки для розв'язання завдань створення і супроводження програмного забезпечення.</p> <p>ПР17. Вміти застосовувати методи компонентної розробки програмного забезпечення.</p> <p>ПР18. Знати та вміти застосовувати інформаційні технології обробки, зберігання та передачі даних.</p> <p>ПР20. Знати підходи щодо оцінки та забезпечення якості програмного забезпечення.</p> <p>ПР23. Вміти документувати та презентувати результати розробки програмного забезпечення.</p>
Ключові слова	обчислення загального призначення на графічних процесорах, general-purpose computing on graphics processing units, GPGPU, графічний процесор, GPU, NVIDIA, CUDA, CUDA C/C++, CUDA Toolkit, ядро CUDA, тензорне ядро, SIMD, OpenCL, CUDA Runtime API, CUDA Driver API, nvcc, BLAS, FFT
Формат курсу	Очний
	Проведення лекцій, лабораторних робіт та консультації для кращого розуміння тем

Теми	Див. Схема курсу
Підсумковий контроль, форма	Іспит у кінці семестру
Пререквізити	Для вивчення цього курсу студентам потрібні базові знання з курсів: <ul style="list-style-type: none"> - Вища математика; - Архітектура комп'ютерів та комп'ютерна електроніка; - Основи програмування; - Вступ в інженерію програмного забезпечення; - Дискретна математика; - Об'єктно-орієнтоване програмування; - Алгоритми і структури даних; - Паралельне програмування; - Методи обчислень; - Операційні системи; - Системне програмування.
Навчальні методи та техніки, які будуть використовуватися під час викладання курсу	Інформаційні методи (лекції, презентації, лабораторні роботи, написання есе, виконання індивідуальних завдань, обговорення, консультації для поглибленого розуміння тем, бесіда, ілюстрація, демонстрація); Дедуктивні методи на основі узагальнень; Евристичні методи (проблемна лекція); Інтерактивні методи (дискусія).
Необхідне обладнання	Для проведення лекційних занять: комп'ютер (мінімальні характеристики: процесор Intel Core i7 (4 ядра / 8 потоків), 16 ГБ оперативної пам'яті, 50 ГБ вільного місця на диску, відеокарта Nvidia GEFORCE GTX 1050 Ti 2048 MB), мультимедійне обладнання (в т.ч. проектор), доступ до мережі Інтернет, Moodle, BigBlueButton. Для проведення лабораторних занять: комп'ютерний клас, комп'ютери (мінімальні характеристики: процесор Intel Core i7 (4 ядра / 8 потоків), 16 ГБ оперативної пам'яті, 50 ГБ вільного місця на диску, відеокарта Nvidia GEFORCE GTX 1050 Ti 2048 MB), GPU-сервер з підтримкою NVIDIA CUDA (мінімальні характеристики: процесор Intel Core i7 (4 ядра / 8 потоків), 32 ГБ оперативної пам'яті, 100 ГБ вільного місця на диску, відеокарта Nvidia GEFORCE GTX 1080 Ti 2048 MB), доступ до мережі Інтернет, Moodle, BigBlueButton. Необхідне програмне забезпечення: CUDA Toolkit, компілятор nvcc, бібліотеки cuRAND, cuBLAS, NPP і Thrust, профілювальник і відладчик gdb (cuda-gdb) для GPU, NVIDIA Nsight IDE або інше інтегроване середовище для розробки мовою CUDA C/C++.
Критерії оцінювання (окремо для кожного виду навчальної діяльності)	Оцінювання проводиться упродовж семестру та під час екзаменаційної сесії за 100-бальною шкалою. Бали нараховуються за такими видами робіт із наступним співвідношенням: <ul style="list-style-type: none"> • лабораторні роботи: 30% семестрової оцінки; максимальна кількість балів – 30. Індивідуальний проект може бути виконаний як альтернатива лабораторним роботам та передбачає розробку комплексного проекту,

який охоплює тематику курсу повною мірою та передбачає розробку програми з використанням технології GPGPU для вирішення складної обчислювальної задачі. Індивідуальний проект оцінюється на основі представлення результатів роботи та проміжних (чорнових) звітів кожного місяця навчального семестру або на основі представлення кінцевих результатів роботи в кінці семестру.

- контрольні заміри (1 модуль): 20% семестрової оцінки; максимальна кількість балів – 20.

Модульне завдання здається впродовж 15-го тижня навчання у формі есе, написаного на тему обчислень загального призначення на графічних процесорах (GPGPU) згідно індивідуального завдання об'ємом 8-10 сторінок друкованого тексту стандартного формату А4.

- екзамен: 50% семестрової оцінки; максимальна кількість балів – 50.

Екзаменаційне завдання здається в письмовому вигляді та складається з трьох завдань: двох теоретичних питань по 15 балів кожне та одного практичного завдання, яке оцінюється в 20 балів.

Підсумкова максимальна кількість за курс – 100 балів.

Академічна доброчесність: Очікується, що лабораторні та контрольні роботи студентів будуть їх оригінальними дослідженнями чи міркуваннями. Відсутність посилань на використані джерела, фабрикування джерел, списування, втручання в роботу інших студентів становлять, але не обмежують, приклади можливої академічної недоброчесності. Виявлення ознак академічної недоброчесності в роботі студента є підставою для її незарахування викладачем, незалежно від масштабів плагіату чи обману.

Відвідання занять є важливою складовою навчання. Очікується, що всі студенти відвідають усі лекції і лабораторні заняття курсу. Студенти мають інформувати викладача про неможливість відвідати заняття. Студенти зобов'язані дотримуватися усіх термінів визначених для виконання усіх видів робіт, передбачених курсом.

Література. Уся література, яку 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.

Політика виставлення балів. Враховуються бали набрані на поточному тестуванні, самостійній роботі та бали підсумкового тестування. При цьому обов'язково враховуються присутність на заняттях та активність студента під час лабораторного заняття; недопустимість пропусків та запізнь на заняття; користування мобільним телефоном, планшетом чи іншими мобільними пристроями під час заняття в цілях не пов'язаних з навчанням; списування та плагіат; несвоєчасне виконання поставленого завдання і т. ін.

Жодні форми порушення академічної доброчесності не толеруються.

Оцінювання лабораторних робіт (10 лабораторних робіт, максимальна кількість балів: 30) відбувається шляхом оцінки роботи студента у вигляді

демонстрації виконаної роботи (0-5 балів за одну роботу) і здачі та захисту звіту по виконаній лабораторній роботі (0-5 балів за одну роботу). У підсумку, всі набрані бали множаться на коефіцієнт 0.3 для переведення у 30-ти бальну шкалу.

Бали оцінювання лабораторних робіт нараховуються за наступним співвідношенням:

5 – студент в повному обсязі володіє навчальним матеріалом, має повне розуміння розглянутої теми, надає правильні відповіді на запитання по темі, код програми функціонує відповідно до завдання;

4 – студент достатньо розуміє розглянутий матеріал та принципи написаного ним коду програми, присутні неточності та незначні помилки у відповідях на запитання по темі, код програми функціонує відповідно до завдання (або з несуттєвими недоліками);

3 – студент не досить добре розуміє розглянутий матеріал та написаний ним код програми, вагається та надає неточні/не конкретні відповіді на запитання по темі, код програми функціонує неточно, або з помірними недоліками;

2 – студент погано розуміє розглянутий матеріал та написаний ним код програми, студент в більшості надає помилкові відповіді на питання по темі, код програми функціонує з суттєвими недоліками;

1 – студент погано розуміє розглянутий матеріал та написаний ним код програми, код програми не функціонує належним чином;

0 – студент зовсім не засвоїв розглянутий матеріал, написаний ним код програми не відповідає темі/не функціонує взагалі.

Оцінювання модульного завдання (1 модуль, максимальна кількість балів: 20) відбувається шляхом написання студентом есе на тему обчислень загального призначення на графічних процесорах (GPGPU) згідно індивідуального завдання об'ємом 8-10 сторінок друкованого тексту стандартного формату А4.

Бали оцінювання модульного завдання нараховуються за наступним співвідношенням:

20-17 – розглянута тема відтворюється в повному обсязі, правильно, обґрунтовано, логічно, які містять аналіз і систематизацію, аргументовані висновки. Засвідчено глибоке володіння матеріалом. Наведені приклади коду повністю робочі та відповідають темі. Можуть бути присутні несуттєві помилки та невідповідності;

17-13 – відтворюється значна частина розглянутої теми. Виявлено знання і розуміння основних положень навчальної дисципліни, проте присутні неточності та/або невідповідності основній темі. Наведені приклади коду частково робочі, проте в загальному відповідають темі;

13-10 – відстежується загальне розуміння розглянутої теми. Виявлені множинні неточності та невідповідності, пояснення наведеного коду відсутні, код функціонує із значними неточностями (або відсутні приклади запуску коду на виконання взагалі);

10-5 – студент погано розуміє розглянуту тему. Виявлені суттєві неточності та невідповідності. Наведені приклади коду з суттєвими недоліками, або не відповідають темі;

	<p>5-0 – студент взагалі не розуміє розглянуту тему. Тему не розкрито, кількість викладеного матеріалу не відповідає загальним нормам обраного виду роботи. Наведений код не робочий, або відсутній як такий.</p> <p>Критерії отримання додаткових балів: Нарахування додаткових балів відбувається за написання тез доповідей, наукових статей, участь у діяльності наукового гуртка, участь у наукових семінарах та круглих столах, конкурсах, участь у заходах неформальної освіти за отримання сертифікатів про проходження навчання на різних освітніх платформах (Coursera, Prometheus тощо), курсах на провідних ІТ компаніях за тематикою навчальної дисципліни. Кількість балів визначається відсотком покриття результатів відповідної активності до вимог результатів навчання з навчальної дисципліни.</p>
Питання до заліку чи екзамену	Перелік питань та завдань для проведення підсумкової оцінки знань певних тем до контрольних робіт розміщені на веб-сторінці курсу.
Опитування	Анкету-оцінку з метою оцінювання якості курсу буде надано по завершенню курсу.

**Схема курсу “Програмування на GPGPU”
для студентів спеціальності 121 – Інженерія програмного забезпечення**

Тиж.	Тема, план, короткі тези	Форма діяльності (заняття) **лекція, самостійна, дискусія, групова робота)	Література. *** Ресурси в інтернеті	Завдання, год	Термін виконання
1	Технологія GPGPU. Реалізація неграфічних обчислень на графічних процесорах	Лекція	[1], [2], [3], [4], [5], [6], Сайт курсу	2	кінець поточного тижня
1	Вступне заняття. Налаштування середовища та необхідного програмного забезпечення.	Лабораторна робота	[6], [7], [8], [9], [12], Сайт курсу	2	кінець поточного тижня
2	Технологія CUDA. Уніфікована обчислювальна архітектура пристроїв	Лекція	[1], [2], [3], [4], [6], [7], [8], [9], [12], Сайт курсу	2	кінець поточного тижня
2	Лабораторна робота №1. Використання системних інструментів для отримання інформації про графічні пристрої	Лабораторна робота	[6], [7], [8], [9], [12], Сайт курсу	2	кінець 7-го тижня
3	Програмний інтерфейс CUDA C/C++. Частина 1.	Лекція	[6], [8], [9], [12], [27], Сайт курсу	2	кінець поточного тижня
3	Лабораторна робота №2. Отримання інформації про графічний пристрій з підтримкою CUDA та вибір активного пристрою	Лабораторна робота	[8], [9], [10], [27], Сайт курсу	2	кінець 8-го тижня
4	Програмний інтерфейс CUDA C/C++. Частина 2.	Лекція	[6], [8], [9], [12], [27], Сайт курсу	2	кінець поточного тижня
4	Лабораторна робота №3. Розпаралелювання задач за допомогою одновимірної структури сітки та блоків	Лабораторна робота	[6], [8], [9], [12], [27], Сайт курсу	2	кінець 9-го тижня
5	Керування пам'яттю в CUDA C/C++. Частина 1.	Лекція	[6], [8], [12], [13], Сайт курсу	2	кінець поточного тижня
5	Лабораторна робота №4. Розпаралелювання задач за допомогою багатовимірної структури сітки та блоків	Лабораторна робота	[6], [8], [9], [12], [27], Сайт курсу	2	кінець 10-го тижня
6	Керування пам'яттю в CUDA C/C++. Частина 2.	Лекція	[6], [8], [12], [13], Сайт курсу	2	кінець поточного тижня
6	Лабораторна робота №5. Заняття 1. Використання різних типів пам'яті CUDA для реалізації фільтрації зображень	Лабораторна робота	[6], [8], [9], [12], [13], [27], Сайт курсу	2	кінець 11-го тижня
7	Потоки та події CUDA	Лекція	[10], [11], [12], [13], Сайт курсу	2	кінець поточного тижня
7	Лабораторна робота №5. Заняття 2. Використання різних типів пам'яті CUDA для реалізації фільтрації зображень	Лабораторна робота	[6], [8], [9], [12], [13], [27], Сайт курсу	2	кінець 11-го тижня
8	Використання CUDA Driver API	Лекція	[12], [13], [14], [27],	2	кінець поточного тижня

			Сайт курсу		тижня
8	Лабораторна робота №6. Заняття 1. Використання потоків та подій CUDA для реалізації асинхронного виконання функцій ядра	Лабораторна робота	[10], [11], [12], [13], [27], Сайт курсу	2	кінець 12-го тижня
9	Бібліотеки CUDA Toolkit. CUDA Math API та cuRAND Library	Лекція	[9], [12], [13], [15], [16], [27], Сайт курсу	2	кінець поточного тижня
9	Лабораторна робота №6. Заняття 2. Використання потоків та подій CUDA для реалізації асинхронного виконання функцій ядра	Лабораторна робота	[10], [11], [12], [13], [27], Сайт курсу	2	кінець 12-го тижня
10	Бібліотеки лінійної алгебри CUDA C/C++: cuBLAS та NVBLAS. Частина 1.	Лекція	[9], [12], [13], [17], [18], [27], Сайт курсу	2	кінець поточного тижня
10	Лабораторна робота №7. Використання бібліотеки cuRAND для оцінки значення числа π методом Монте-Карло	Лабораторна робота	[9], [12], [13], [16], [27], Сайт курсу	2	кінець 13-го тижня
11	Бібліотеки лінійної алгебри CUDA C/C++: cuBLAS та NVBLAS. Частина 2.	Лекція	[9], [12], [13], [17], [18], [27], Сайт курсу	2	кінець поточного тижня
11	Лабораторна робота №8. Використання бібліотеки cuBLAS для множення матриць	Лабораторна робота	[9], [12], [13], [17], [27], Сайт курсу	2	кінець 14-го тижня
12	Високорівнева алгоритмічна бібліотека Thrust	Лекція	[9], [12], [13], [19], [20], [27], Сайт курсу	2	кінець поточного тижня
12	Лабораторна робота №9. Використання бібліотеки Thrust для роботи з векторами даних	Лабораторна робота	[9], [12], [13], [19], [20], [27], Сайт курсу	2	кінець 15-го тижня
13	Бібліотека обробки цифрових сигналів та зображень NPP	Лекція	[9], [12], [13], [21], [22], [27], Сайт курсу	2	кінець поточного тижня
13	Лабораторна робота №10. Заняття 1. Фільтрація та обробка зображень за допомогою бібліотеки NPP	Лабораторна робота	[9], [12], [13], [21], [22], [27], Сайт курсу	2	кінець 16-го тижня
14	Бібліотеки швидкого перетворення Фур'є: cuFFT та cuFFTDx	Лекція	[9], [12], [13], [23], [24], [27], Сайт курсу	2	кінець поточного тижня
14	Лабораторна робота №10. Заняття 2. Фільтрація та обробка зображень за допомогою бібліотеки NPP	Лабораторна робота	[9], [12], [13], [21], [22], [27], Сайт курсу	2	кінець 16-го тижня
15	Модульне заняття	Лекція	Сайт курсу	2	кінець поточного тижня
15	Захисне лабораторне заняття	Лабораторна робота	Сайт курсу	2	кінець поточного тижня
16	Підсумкове заняття	Лекція	Сайт курсу	2	кінець поточного тижня
16	Захисне лабораторне заняття	Лабораторна робота	Сайт курсу	2	кінець поточного тижня