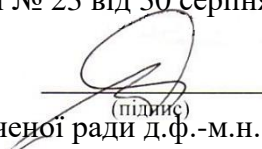


МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Львівський національний університет імені Івана Франка
Факультет прикладної математики та інформатики
Кафедра дискретного аналізу та інтелектуальних систем
Кафедра інформаційних систем
Кафедра програмування

Затверджено

на засіданні Вченої ради
факультету прикладної математики та
інформатики
Львівського національного університету
імені Івана Франка
(протокол № 23 від 30 серпня 2023 р.)


(підпис)
Голова Вченої ради д.ф.-м.н., проф. Дияк І.І.

Силабус

«Навчальної (обчислювальної) практики»,
що проводиться в межах ОПП Інформатика
першого (бакалаврського) рівня вищої освіти
для здобувачів за спеціальністю 122 Комп'ютерні науки

Львів 2023 р.

Назва дисципліни	Навчальна (обчислювальна) практика
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 дискретного аналізу та інтелектуальних систем Кафедра інформаційних систем Кафедра програмування
Галузь знань, шифр та назва спеціальності	Галузь знань: 12 Інформаційні технології Спеціальність: 122 Комп'ютерні науки
Викладачі дисципліни	Смичок Марія Богданівна, асистент кафедри дискретного аналізу та інтелектуальних систем Козій Ірина Ярославівна, доцент кафедри інформаційних систем Ярошко Оксана Сергіївна, асистент кафедри інформаційних систем Галамага Любомир Богданович, асистент кафедри програмування
Контактна інформація викладачів	Електронна пошта: maria.smychok@lnu.edu.ua , iryna.kozij@lnu.edu.ua , oksana.yaroshko@lnu.edu.ua , liubomyr.halamaha@lnu.edu.ua веб-сторінки: https://ami.lnu.edu.ua/employee/smychok-mariya , https://ami.lnu.edu.ua/employee/kozij-i-ya , https://ami.lnu.edu.ua/employee/yaroshko-o-s , https://ami.lnu.edu.ua/employee/halamaha
Консультації з питань навчання по дисципліні відбуваються	Консультації проводять раз на тиждень згідно з оприлюдненим розкладом консультацій викладача. Можливі он-лайн консультації через Zoom чи Microsoft Teams. Для погодження часу он-лайн консультацій слід писати на електронну пошту викладача.
Сторінка курсу	https://ami.lnu.edu.ua/course/educational-practice-programming
Інформація про дисципліну	«Навчальна (обчислювальна) практика» входить до циклу професійної і практичної підготовки зі спеціальності 122 Комп'ютерні науки для освітньо-професійної програми «Інформатика», яку проводять у першому і другому семестрах в обсязі 3 кредити (за Європейською кредитно-трансферною системою ECTS) Практика відбувається без відриву від навчання упродовж навчального року в комп'ютерних лабораторіях університету по одній парі в тиждень.
Коротка анотація дисципліни	Фокус уваги практики спрямовано на те, щоб навчити студентів застосовувати знання та вміння, здобуті в курсі «Програмування», використовувати середовище програмування для самостійного написання, тестування та налагодження програм, зберігання програмних проектів у віддаленому сховищі. Студенти виконують індивідуальні та командні завдання, звітують про виконану роботу. Завдання стосуються процедурного, об'єктно-орієнтованого та узагальненого програмування.
Мета та цілі дисципліни	Метою «Навчальної (обчислювальної) практики» є практична підготовка здобувача освіти до створення програмних продуктів. Вона повинна навчити студента: <ul style="list-style-type: none"> створювати програмні проекти мовою C++ за допомогою середовища програмування, виправляти синтаксичні та логічні помилки, виконувати покрокове виконання програм, модульне тестування; використовувати технологію git і хмарні сервіси, наприклад, GitHub для створення локального та віддаленого репозиторію, виконання індивідуальних і командних завдань; планувати свою роботу для вчасного виконання завдань, розподіляти обов'язки в команді, узгоджувати власні дії з іншими членами команди під час виконання командних завдань; застосовувати процедурну методологію та об'єктно-орієнтований підхід до проектування і написання програм; застосовувати принципи узагальненого програмування для побудови та використання шаблонів функцій та класів, для метапрограмування;

	<ul style="list-style-type: none"> використовувати узагальнені алгоритми та контейнери стандартної бібліотеки STL для написання власних програм, доповнювати можливості бібліотеки власними шаблонами.
Література для вивчення дисципліни	<p><i>Основна література</i></p> <ol style="list-style-type: none"> Ярошко С. А. Методи розробки алгоритмів. Програмування мовою C++: навч. посібник / С.А. Ярошко, О.С. Ярошко – Львів: ЛНУ імені Івана Франка, 2022. – 248 с. – https://lnuittutor.github.io/ Бублик В.В. Об'єктно-орієнтоване програмування: [Підручник] / В.В. Бублик. – К.: ІТ-книга, 2015. – 624 с. Stephen Prata C++ Primer Plus 6th Edition (Developer's Library) In 2 Volumes – Addison-Wesley Professional, 2011. – 1440 p. Дудзяний І.М. Програмування мовою C++. Частина 1: Парадигма процедурного програмування: навчальний посібник / І.М. Дудзяний. – Львів: ЛНУ імені Івана Франка, 2013. – 468 с. Microsoft Learn: C++ language documentation. – Електронний ресурс. Режим доступу: https://learn.microsoft.com/en-us/cpp/cpp/?view=msvc-170 https://git-scm.com/book/uk/v2 <p><i>Додаткова література</i></p> <ol style="list-style-type: none"> Bruce Eckel Thinking in C++, Vol. 1: Introduction to Standard C++, 2nd Edition – Prentice Hall, 2000. – 814 p. Bruce Eckel Thinking in C++, Vol. 2: Practical Programming, 2nd Edition – Prentice Hall, 2003. – 832 p. Scott Meyers Effective Modern C++ – O'Reilly Media, 2015. – 316 p. Nicolai M. Josuttis The C++ Standard Library. A tutorial and Reference. Second Edition – Addison-Wesley, 2012. – 1162 p. http://www.cplusplus.com/ https://www.geeksforgeeks.org/c-plus-plus/ https://metanit.com/cpp/ https://www.codeproject.com/
Обсяг курсу	3 кредити ЄКТС – 90 годин. З них 90 год. самостійної роботи.
Очікувані результати навчання	<p>Після завершення цього курсу студент буде:</p> <p><i>знати</i></p> <ul style="list-style-type: none"> засоби та можливості середовища програмування, зокрема, щодо створення програмних проєктів, тестування та налагодження програм; засоби створення локальних і віддалених сховищ; синтаксис мови програмування C++, склад і правила використання бібліотеки STL; засади модульного, об'єктно-орієнтованого та узагальненого програмування; призначення та влаштування базових структур даних і алгоритмів для їхнього опрацювання. <p><i>вміти</i></p> <ul style="list-style-type: none"> використовувати інтегроване середовище програмування, хмарні сховища репозиторіїв програмного коду для зберігання та поширення власних програм, для роботи в команді. писати та налагоджувати консольні застосунки мовою C++; виконувати модульне тестування бібліотек функцій та класів; проектувати, оголошувати та використовувати ієрархії класів, шаблони класів; використовувати та доповнювати засоби бібліотеки STL.
Компетентності	<p><i>Інтегральна:</i> Здатність розв'язувати складні спеціалізовані задачі та практичні проблеми у галузі комп'ютерних наук або у процесі навчання, що передбачають застосування теорій та методів інформаційних технологій і характеризуються комплексністю та невизначеністю умов.</p> <p><i>Загальні (ЗК):</i></p> <p>ЗК1. Здатність до абстрактного мислення, аналізу та синтезу.</p> <p>ЗК2. Здатність застосовувати знання у практичних ситуаціях.</p> <p>ЗК3. Знання та розуміння предметної області та розуміння професійної діяльності.</p> <p>ЗК7. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.</p> <p>ЗК8. Здатність генерувати нові ідеї (креативність).</p> <p>ЗК9. Здатність працювати в команді.</p> <p>ЗК12. Здатність оцінювати та забезпечувати якість виконуваних робіт.</p>

	<p><i>Спеціальні (фахові, предметні) компетентності (СК):</i></p> <p>СК3. Здатність до логічного мислення, побудови логічних висновків, використання формальних мов і моделей алгоритмічних обчислень, проектування, розроблення й аналізу алгоритмів, оцінювання їх ефективності та складності, розв'язності та нерозв'язності алгоритмічних проблем для адекватного моделювання предметних областей і створення програмних та інформаційних систем.</p> <p>СК8. Здатність проектувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: узагальненого, об'єктно-орієнтованого, функціонального, логічного, з відповідними моделями, методами й алгоритмами обчислень, структурами даних і механізмами управління.</p> <p>СК10. Здатність застосовувати методології, технології та інструментальні засоби для управління процесами життєвого циклу інформаційних і програмних систем, продуктів і сервісів інформаційних технологій відповідно до вимог замовника.</p>
Програмні результати навчання	<p>ПР9. Розробляти програмні моделі предметних середовищ, вибирати парадигму програмування з позицій зручності та якості застосування для реалізації методів та алгоритмів розв'язання задач в галузі комп'ютерних наук.</p> <p>ПР14. Знати мережні технології, архітектури комп'ютерних мереж, мати практичні навички технології адміністрування комп'ютерних мереж та їх програмного забезпечення.</p> <p>ПР15. Застосовувати знання методології та CASE-засобів проектування складних систем, методів структурного аналізу систем, об'єктно-орієнтованої методології проектування при розробці і дослідженні функціональних моделей організаційно-економічних і виробничо-технічних систем.</p>
Ключові слова	Алгоритм, програма мовою С++, тип даних, інструкція галуження, інструкція повторення, масив, вказівник, функція, клас, об'єкт, наслідування, поліморфізм, віртуальний метод, шаблон класу, контейнер, функтор, репозиторій, покрокове виконання, модульне тестування.
Формат курсу	Очний.
Теми	<p style="text-align: center;">1 семестр</p> <p>Вступ. Ознайомлення з головними завданнями практики.</p> <p><i>Індивідуальні завдання</i></p> <ol style="list-style-type: none"> 1. Реалізація обчислень за математичними формулами (послідовні алгоритми), наочне введення, виведення. – 1-й тиждень, 4 бали. 2. Програмна реалізація алгоритмів з розгалуженнями. – 2-й тиждень, 3 бали. 3. Побудова арифметичних й ітераційних циклів (задачі цілочислової арифметики, опрацювання послідовностей). – 3-й тиждень, 4 бали. 4. Побудова, трансформація одно- та багатовимірних масивів, у тому числі динамічних. – 4-й тиждень, 4 бали. 5. Опрацювання рядків. – 5-й тиждень, 3 бали. 6. Визначення та використання простих функцій. Рекурсія. – 6-й – 7-й тижні, 5 балів. 7. Визначення та використання функцій вищого порядку. – 8-й тиждень, 4 бали. 8. Визначення та використання простих структур, перевизначення операторів. – 9-й – 10-й тижні, 5 балів. 9. Використання файлів різних форматів для зберігання та завантаження даних. – 11-й тиждень, 4 бали. 10. Опрацювання виняткових ситуацій. – 12-й тиждень, 4 бали. <p><i>Командні завдання</i></p> <ol style="list-style-type: none"> 11. Моделювання зв'язних структур: одно- та двозв'язних списків, бінарних дерев, дерев пошуку. – 13-й – 14-й тижні, 5 балів. 12. Створення найпростіших класів і об'єктів. – 15-й – 16-й тижні, 5 балів. <p style="text-align: center;">2 семестр</p> <p><i>Командні завдання</i></p>

	<p>13. Розробка ієрархії класів. Просте наслідування класів, створення поліморфного масиву екземплярів. – 1-й – 2-й тижні, 5 балів.</p> <p>14. Побудова класів-комполітів. – 3-й – 4-й тижні, 5 балів.</p> <p>15. Розробка власного класу-контейнера з багатими функціональними можливостями. – 5-й – 6-й тижні, 5 балів.</p> <p><i>Індивідуальні завдання</i></p> <p>16. Випробування множинного, закритого, захищеного наслідування. – 7-й тиждень, 5 балів.</p> <p>17. Розробка та використання шаблону функції. – 8-й тиждень, 5 балів.</p> <p>18. Розробка та використання шаблону класу. – 9-й тиждень, 5 балів.</p> <p>19. Використання послідовних контейнерів і алгоритмів бібліотеки STL для створення/опрацювання колекції об'єктів. – 10-й – 11-й тижні, 5 балів.</p> <p>20. Використання асоціативних контейнерів і алгоритмів бібліотеки STL для створення/опрацювання колекції об'єктів. – 12-й – 13-й тижні, 5 балів.</p> <p>21. Оголошення та використання класів характеристик для налаштування шаблону функції/класу. – 14-й тиждень, 5 балів.</p> <p>22. Серіалізація/десеріалізація об'єктів. – 15-й – 16-й тижні, 5 балів.</p>
Підсумковий контроль, форма	Диференційований залік у кінці другого семестру.
Пререквізити	Для проходження практики студенти потребують базових знань з курсів «Програмування»; «Алгебра та геометрія»; «Дискретна математика»; «Алгоритми і структури даних», достатніх для розроблення застосунків мовою C++.
Навчальні методи та техніки, які використовують під час викладання курсу	Навчальна (обчислювальна) практика покликана виявити та розвинути індивідуальні здібності здобувача освіти щодо створення програм мовою C++, тому вона передбачає здебільшого самостійне виконання завдань. Студент пише програми, перевіряє їхню правильність, оформляє звіт про виконане завдання і захищає його. Керівник практики надає консультації щодо формулювання завдань, допомагає з правильним налаштуванням перших програмних проєктів, сховищ. Тренує зі здобувачами правильну послідовність дій щодо покрокового виконання програм, створення тестів тощо. Керівник організує колективні обговорення командних завдань, допомагає з розподілом обов'язків і презентацією отриманих результатів.
Необхідне обладнання	Комп'ютер, ОС Windows/Linux, доступ до інтернету, середовище програмування мовою C++ (Microsoft Visual Studio, Code Blocks тощо). 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.
Критерії оцінювання (окремо для кожного виду навчальної діяльності)	<p>Оцінювання виконують за 100-бальною шкалою: 50 балів за виконані завдання у першому семестрі, 50 балів – у другому. Звіт про практику можна оформити як окремий документ, або як належним чином оформлене сховище (сховища) на GitHub. Опис кожного завдання повинен містити умову, текст програми з коментарями, аналіз отриманих результатів, модульні тести.</p> <p>Додаткові бали будуть зараховані учасникам Всеукраїнської студентської олімпіади з програмування (відповідно до кількості розв'язаних задач).</p> <p>Відвідання занять є важливою складовою навчання. Очікується, що всі студенти відвідають усі заняття практики. Активність під час занять заохочується балами. У будь-якому випадку студенти зобов'язані дотримуватися усіх строків виконання завдань практики. Виконані роботи завантажують у відповідне хмарне сховище. Якщо заняття за наказом ректора відбуваються дистанційно, студенти повинні долучатися до онлайн-нарад з увімкненими відеореєстраторами.</p> <p>Академічна доброчесність: очікується, що роботи студентів будуть їхнім оригінальними дослідженнями чи міркуваннями. Відсутність посилань на</p>

	використані джерела, фабрикування джерел, списування, втручання в роботу інших студентів, здавання чужих комп'ютерних програм як своїх становлять, але не обмежують, приклади можливої академічної недоброчесності. Виявлення ознак академічної недоброчесності в письмовій роботі студента є підставою для її незарахування викладачем, незалежно від масштабів плагіату чи обману.
Опитування	Анкету-оцінку з метою оцінювання якості курсу буде надано після завершення курсу.