

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

Затверджено

На засіданні
кафедри оптоелектроніки та інформаційних
технологій
факультету електроніки та комп'ютерних
технологій
Львівського національного університету імені
Івана Франка
(протокол № 6 від 29.09. 2023 р.)

Завідувач кафедри:



Олег КУШНІР

Силабус з навчальної дисципліни
«Основи програмування»,
що викладається в межах ОПП «Інженерія програмного
забезпечення» першого (бакалаврського) рівня вищої освіти для
здобувачів зі спеціальності
121 «Інженерія програмного забезпечення»

Львів 2023 р.

Назва дисципліни	Основи програмування
Адреса викладання дисципліни	м. Львів, вул. Драгоманова 50, вул. Ген. Тарнавського 107.
Факультет та кафедра, за якою закріплена дисципліна	Факультет електроніки та комп'ютерних технологій, кафедра оптоелектроніки та інформаційних технологій
Галузь знань, шифр та назва спеціальності	12 Інформаційні технології, 121 Інженерія програмного забезпечення
Викладачі дисципліни	Крупич Олег Миколайович, доктор фіз.-мат. наук, ст.н.сп., професор; Катеринчук Іван Миколайович, канд. фіз.-мат. наук, доцент, доцент; Дуфанець Марта Василівна, канд. фіз.-мат. наук, доцент, доцент.
Контактна інформація викладачів	oleh.kranych@lnu.edu.ua, https://electronics.lnu.edu.ua/employee/kranych-oleh-mykolayovych ; ivan.katerynychuk@lnu.edu.ua, https://electronics.lnu.edu.ua/employee/katerynychuk-ivan-mykolajovych ; marta.dufanets@lnu.edu.ua, https://electronics.lnu.edu.ua/employee/dufanets-marta-vasylivna
Консультації з питань навчання по дисципліні	Консультації в день проведення лекційних занять (за попередньою домовленістю). Також можливі он-лайн консультації через MS Teams. Для погодження часу он-лайн консультацій слід писати на електронну пошту викладача.
Сторінка дисципліни	https://electronics.lnu.edu.ua/course/osnovy-prohramuvannia-ipz
Інформація про дисципліну	Дисципліна «Основи програмування» є нормативною дисципліною зі спеціальності 121 Інженерія програмного забезпечення для освітньої програми «Інженерія програмного забезпечення», яка викладається в 1 семестрі в обсязі 5,0 кредитів (за Європейською Кредитно-Трансферною Системою ECTS).
Коротка анотація дисципліни	Навчальну дисципліну розроблено таким чином, щоб ознайомити студентів з базовими поняттями про: основи програмування, абстракцію та алгоритми, компіляцію та інтерпретацію, середовище розробки, ключові слова, ідентифікатори та модифікатори, оператори, оператори галуження та цикли, функції, масиви одномірні та багатомірні, стрічки та роботу з ними, структуру пам'яті Сі програми, вказівники, структури, систему вводу / виводу, робота з файловою системою, директиви.
Мета та цілі дисципліни	Метою вивчення нормативної дисципліни «Основи програмування» є одержання студентами необхідних теоретичних та практичних знань з алгоритмізації та програмування, а також формування в студентів практичних навичок, які б дали змогу ефективно застосовувати знання з основ програмування в подальшій роботі та навчальному процесі.
Література для вивчення дисципліни	Основна література: 1. ISO/IEC 9899:2018 [ISO/IEC 9899:2018] Information technology — Programming languages — C 2. Брайан В. Керніган, Деніс М. Річі. Мова програмування C. Друге видання, 2015. – 232 с. http://programming.in.ua/programming/c-language/227-book-programming-c-kernighan.html 3. Хвищун І.О. Алгоритмізація та програмування: Лекції. Матеріали для самостійної роботи. – Львів: Видавничий центр ЛНУ імені Івана Франка, 2017. –278с.

	<p>4. Злобін Г.Г. Основи алгоритмізації та програмування мовою Сі. Підручник. – Київ: «Каравела», 2022. – 168 с.</p> <p>Додаткова література:</p> <p>5. David M. Harris, Sarah L. Harris. Digital Design and Computer Architecture. Second Edition. Elsevier, Inc. -2013. -690 p.</p> <p>6. Вишневий С.В., Катін П.Ю., Крилов Є.В. Інформатика. Основи програмування та алгоритми. Мова програмування С. Лабораторний практикум. – Київ : КПІ ім. Ігоря Сікорського, 2022. – 221 с.</p> <p>7. Daniel Zingaro. Algorithmic Thinking: A Problem-Based Introduction. No Starch Press. – 2020. – 408 p.</p> <p>8. Robert Seacord. Effective C: An Introduction to Professional C Programming. No Starch Press. – 2020. – 272 p.</p> <p>9. Anthony Wallit. Learning C programming: An Informative and In-depth Guide to C Programming. Independently published. – 2022. – 250 p.</p> <p>10. Jeff Szuhay. Learn C Programming: A beginner's guide to learning the most powerful and general-purpose programming language with ease, 2nd Edition. Packt Publishing. – 2022. – 742 p.</p>
<p>Обсяг курсу</p>	<p>64 години аудиторних занять. З них 32 години лекцій, 32 години лабораторних робіт та 86 годин самостійної роботи (всього 150 год.)</p>
<p>Очікувані результати навчання</p>	<p>Після завершення цього курсу студент буде:</p> <ul style="list-style-type: none"> – Володіти здатністю до абстрактного мислення, аналізу та синтезу, а також до алгоритмічного та логічного мислення; знати актуальність вивчення алгоритмів та мов програмування, методи представлення алгоритмів, способи програмної реалізації алгоритмів та засоби відлагодження програм. – Вміти класифікувати та розв'язувати складні спеціалізовані завдання інженерії програмного забезпечення, пов'язані зі створенням та програмною реалізацією алгоритмів, вміти застосовувати знання основ програмування в процесі практичного написання програмного коду, використовувати сучасні середовища розробки під час створення та відлагодження програм різного призначення. <p>Після вивчення даного курсу «Веб-технології та програмування (ч.1)» здобувачі набувають таких Загальних та Фахових компетентностей та Програмних результатів навчання:</p> <p>ЗК01. Здатність до абстрактного мислення, аналізу та синтезу.</p> <p>ЗК02. Здатність застосовувати знання у практичних ситуаціях.</p> <p>ФК15. Здатність розробляти архітектури, модулі та компоненти програмних систем.</p> <p>ФК23. Здатність реалізовувати фази та ітерації життєвого циклу програмних систем та інформаційних технологій на основі відповідних моделей і підходів розробки 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>ФК29. Здатність здійснювати розробку програмного забезпечення використовуючи різні методології програмування (в тому числі паралельне, об'єктно-орієнтоване, функціональне програмування та інші)</p> <p>ПРН05. Знати і застосовувати відповідні математичні поняття, методи доменного, системного і об'єктно-орієнтованого аналізу та математичного моделювання для розробки програмного забезпечення.</p>

	<p>ПРН07. Знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення.</p> <p>ПРН08. Вміти розробляти людино-машинний інтерфейс.</p> <p>ПРН13. Знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань.</p> <p>ПРН17. Вміти застосовувати методи компонентної розробки програмного забезпечення.</p>
Ключові слова	Абстракція, алгоритм, компіляція, інтерпретація, середовище розробки, типи даних, ідентифікатор, оператор, цикл, функція, масив, стрічка, пам'ять, вказівник, структура, ввід, вивід, файл, директива.
Формат курсу	Очний, або дистанційний (заняття в MS Teams), або комбінований. Проведення лекцій, лабораторних робіт та консультацій для кращого розуміння тем.
Теми	Див. СХЕМА КУРСУ.
Підсумковий контроль, форма	Іспит в кінці семестру.
Пререквізити	Для вивчення курсу студенти потребують базових знань, одержаних при здобутті середньої освіти, насамперед з предметів «Інформатика», «Математика», «Фізика» та ін.
Навчальні методи та техніки, які будуть використовуватися під час викладання курсу	Презентація, лекції, лабораторні роботи, обговорення, дискусія.
Необхідне обладнання	<p>Для проведення лекційних занять:</p> <ul style="list-style-type: none"> • Ноутбук: дисплей 15.6", процесор Intel i7-1065G7, 12 GB оперативної пам'яті, SSD 256 GB (або краще); • мультимедійне обладнання (проектор, проекційний екран, дошка настінна, звуковий підсилювач та аудіосистема); • комутатор мережевий для доступу до мережі Internet. <p>Для проведення лабораторних занять:</p> <ul style="list-style-type: none"> • комп'ютерна лабораторія з 12-14 робочими місцями; • монітори TFT 23"; • системні блоки (процесор Intel i5-6500, 8GB оперативної пам'яті, HDD 256GB); • мультимедійне обладнання (проектор, проекційний екран, дошка настінна, звуковий підсилювач та аудіосистема); • комутатор мережевий для доступу до мережі Internet. <p>Необхідне програмне забезпечення:</p> <ul style="list-style-type: none"> • операційна система ОС Ubuntu 16 LTS; • середовище розробки на мові програмування C: Code::Blocks (версії 17.12); • компілятор мови програмування C (пакет GCC версії 11.2); • SFTP клієнт FileZilla версії 3.57.0; • платформи Moodle та MS Teams.
Критерії оцінювання (окремо для кожного)	Оцінювання проводиться упродовж семестру за 100-бальною шкалою. Бали нараховуються за такими видами робіт з наступним співвідношенням:

виду навчальної діяльності)

- лабораторні роботи: 36% семестрової оцінки; максимальна кількість балів 36;
- контрольні заміри (2 модулі по 7 балів): 14% семестрової оцінки; максимальна кількість балів 14;
- іспит: 50% семестрової оцінки; максимальна кількість балів 50.

Підсумкова максимальна кількість балів 100.

Оцінювання лабораторних робіт (12 лабораторних робіт, максимальна кількість балів 36) відбувається шляхом оцінки теоретичної підготовки студента до лабораторної роботи (0-5 балів за одну роботу) та захисту звіту по виконаній роботі (0-5 балів за одну роботу). Загальна оцінка за лабораторну роботу дорівнює сумі балів (до 10), помноженій на коефіцієнт 0,333.

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.

Контрольні заміри проводяться у формі тесту, який містить 7 питань для кожного студента, по 1 балу за кожне питання.

Бали оцінювання контрольних замірів нараховуються за наступною схемою:

7 балів – студент правильно відповів на всі 7 питань тесту;

6 балів – студент правильно відповів на 6 питань тесту;

5 балів – студент правильно відповів на 5 питань тесту;

4 балів – студент правильно відповів на 4 питання тесту;

3 балів – студент правильно відповів на 3 питання тесту;

2 балів – студент правильно відповів на 2 питання тесту;

1 балів – студент правильно відповів на 1 питання тесту;

0 балів – студент відповів неправильно на всі 7 питань тесту.

Оцінка за іспит (максимально 50 балів) складається із оцінки за відповіді на питання екзаменаційного білета (5 питань), в т.ч., за теоретичні питання (4 питання по 5 балів – максимально 20 балів) і оцінки за практичне завдання (створення і демонстрація роботи програми) – до 20 балів, а також оцінки за співбесіду (2 додаткові питання по матеріалу курсу) – до 10 балів.

Бали оцінювання теоретичних питань (в т.ч. додаткових питань в межах

співбесіди) нараховуються за наступним співвідношенням:

5 – студент в повному обсязі володіє навчальним матеріалом, має повне розуміння розглянутого питання, надає правильні відповіді на запитання по темі, подає правильні приклади програмної реалізації теоретичного питання;

4 – студент достатньо розуміє розглянутий матеріал та принципи написання відповідного коду програми, присутні неточності та незначні помилки у відповідях на запитання, подає в цілому правильні приклади програмної реалізації питання, але вони є неповними або містять несуттєві недоліки;

3 – студент не досить добре розуміє розглянутий матеріал та написання на його основі коду програми, вагається та надає неточні/не конкретні відповіді на запитання по темі, приклади програмної реалізації є неточними, або з помірними недоліками;

2 – студент погано розуміє розглянутий матеріал та написання на його основі коду програми, студент в більшості надає помилкові відповіді на питання по темі, приклади програмної реалізації мають суттєві недоліки;

1 – студент погано розуміє розглянутий матеріал та написання на його основі коду програми, приклади програмної реалізації є фрагментарними і містять принципові недоліки;

0 студент зовсім не засвоїв розглянутий матеріал, не може написати на його основі коду програми, не може навести приклади його програмної реалізації.

Оцінка за співбесіду дорівнює сумі балів за відповіді на 2 додаткові питання по матеріалу курсу.

Бали оцінювання практичного завдання нараховуються за наступними критеріями:

- 1) алгоритм реалізації завдання – до 5 балів;
- 2) працездатність програми, використання ресурсів комп'ютера – до 5 балів;
- 3) інтерфейс користувача, повнота і зручність вводу даних та виводу результатів – до 5 балів;
- 4) оптимальність коду, використання циклів та функцій, наявність коментарів – до 5 балів.

Академічна доброчесність: Очікується, що лабораторні та контрольні роботи студентів будуть їхніми 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. Жодні форми порушення академічної доброчесності не толеруються.

Відвідання занять є важливою складовою навчання. Очікується, що всі студенти відвідають усі лекції і лабораторні заняття курсу. Студенти повинні 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

	<p>третім особам. Студенти заохочуються до використання також й іншої літератури та джерел, яких немає серед рекомендованих.</p> <p>Політика виставлення балів. Враховуються бали набрані на поточному контролі, бали за виконання лабораторних робіт та бали підсумкового іспиту. При цьому обов'язково враховуються присутність на заняттях та активність студента під час лабораторних занять; 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.</p>
Питання до контрольних робіт	Перелік питань та завдань для проведення підсумкової оцінки знань будуть розміщені на веб-сторінці курсу.
Опитування	Анкету-оцінку з метою оцінювання якості курсу буде надано по завершенні курсу.

СХЕМА КУРСУ

Тиж.	Тема, план, короткі тези	Форма діяльності (заняття)	Література. Ресурси в Інтернеті	Завдання (лабораторна робота), год	Термін виконання
1	Вступ до курсу основи програмування. Історичні аспекти розвитку теоретичних та апаратних засобів для здійснення обчислень. Поняття про комп'ютер, інформацію, дані. Зміст етапів інформаційного процесу. Інформатика та комп'ютерні науки. Історія виникнення мови Сі. Місце Сі серед інших мов програмування.	Лекція	3, 4, 6		
2	Абстракція та алгоритми. Поняття абстракції. Двійкова/шістнадцяткова системи числення. ASCII таблиця. Принципи побудови зображення. Поняття алгоритму та його ефективності. Алгоритм половинного ділення. Способи представлення алгоритмів. Псевдокод.	Лекція	3, 4, 6, 7	Основи алгоритмізації. Найпростіші алгоритми. 2 год.	3 тиж. семестру
3	Компіляція та інтерпретація. Середовище розробки. Трансляція програм. Компілятори та інтерпретатори. Принципи побудови та структура проекту. Середовища розробки Scratch, Visual Studio, CodeBlock. Онлайн-розробка. Засоби відлагодження коду.	Лекція	3, 4, 6	Графічна мова програмування Scratch. 2 год.	4 тиж. семестру
4	Мова програмування Сі. Ключові слова, ідентифікатори та модифікатори. Складові частини проекту на Сі. Оператори і операнди. Змінні, оголошення змінних. Ідентифікатори. Ключові слова. Типи даних. Модифікатори типів. Модифікатори доступу. Локальні і глобальні змінні, область видимості змінної. Специфікатори зберігання.	Лекція	1, 2, 5, 6, 9, 10	Мова програмування Сі, компіляція, функції вводу/виводу. 2 год.	5 тиж. семестру
5, 6	Оператори. Оператор присвоювання та ініціалізація змінних. Перетворення типів при присвоюванні. Константи, символічні константи з зворотним слешем. Арифметичні оператори. Оператори відношення та логічні оператори. Побітові оператори. Оператор ?. Оператори & та *. Оператор sizeof. Оператор ,, Оператори . та ->. Оператори [] та (). Пріоритети операторів. Приведення типів компілятором. Примусове перетворення типів. Пробіли і дужки. Скорочення операторів.	Лекція	1, 2, 5, 8, 9, 10	Основні типи змінних. Операції та їх пріоритет. 4 год.	7 тиж. семестру
7	Оператори галуження та цикли. Поняття галуження за умовою, оператори: if і switch. Цикли: while, for і do/while. Оператори переходу: break, continue. Конструкція if-else-if. Розв'язок рівняння алгоритмом половинного ділення.	Лекція	1, 2, 5, 6, 8, 10	Умовні оператори. 2 год. Цикли. 2 год.	9 тиж. семестру
8	Функції. Функції, призначення, оголошення. Попереднє оголошення функції. Оголошення функції через прототип. Прототип функції з невідомим числом параметрів та без параметрів. Правила видимості функцій. Аргументи функцій та передача аргументу через значення або вказівник. Рекурсія. Вказівник на функцію.	Лекція	1, 2, 6, 8, 10	Функції. 4 год	10 тиж. семестру

9,10	Масиви, стрічки та робота з ними. Одномірні масиви. Багатомірні масиви. Передача масивів в функцію. Алгоритми сортування масивів Стрічки та стрічкові константи. Функції для роботи зі стрічками. Масиви стрічок. Індексування з допомогою вказівників. Динамічні масиви. Ініціалізація розмірного та безрозмірного масиву.	Лекція	1, 2, 5, 8, 9, 10	Масиви. 2 год. Стрічки 2 год.	12 тиж. семестру
11, 12	Структура пам'яті Сі програми. Вказівники. Структура пам'яті Сі програми. Стек. Купа. Динамічне виділення пам'яті і вказівники. Вказівники. Вказівники на константи. Вказівники і масиви. Вказівники на символічні масиви. Масиви вказівників. Вказівники на вказівники. Вказівники на функції.	Лекція	1, 2, 5, 8, 9	Вказівники. 4 год.	13 тиж. семестру
13	Структури. Поняття про структури. Доступ до полів структури. Присвоєння структур. Масиви структур. Передача полів структури в функцію. Вказівники на структури. Масиви і структури в структурах. Вкладені структури. Бітові поля. Об'єднання. Переліки. Оператор typedef.	Лекція	1, 2, 5, 6, 8, 10	Структури. 2 год.	14 тиж. семестру
14	Система вводу / виводу. Система вводу / виводу. Текстові потоки. Двійкові потоки. Файли робота з файлами в Сі. Консольний ввід/вивід.	Лекція	1, 2, 6, 8, 10	Мова програмування Сі, компіляція, функції вводу/виводу. 2 год.	15 тиж. семестру
15	Робота з файловою системою. Функції файлової системи ANSI C. Вказівник на файл. Відкриття файлу. Читання і запис файлу. Закриття файлу.	Лекція	1, 2, 5, 6, 8, 10	Робота з файлами. 2 год.	16 тиж. семестру
16	Директиви. Препроцесор, компіляція, директиви. Директиви #define та #undef. Оператори «#» та «##». Директиви умовної компіляції. Директиви #include, #error та #pragma. Оператор «defined». Наперед визначені макроси Сі..	Лекція	1, 2, 5, 8, 9		