



ОСНОВИ ПРОГРАМУВАННЯ. Частина 1. БАЗОВІ КОНСТРУКЦІЇ

Силабус освітнього компонента

Реквізити навчальної дисципліни

Рівень вищої освіти	Перший (бакалаврський)
Галузь знань	F Інформаційні технології
Спеціальність	F2 Інженерія програмного забезпечення
Освітня програма	Інженерія програмного забезпечення інтелектуальних кібер-фізичних систем в енергетиці
Статус дисципліни	Обов'язкова (нормативна)
Форма навчання	Очна (денна)
Рік підготовки, семестр	I курс, перший (осінній) семестр
Обсяг дисципліни	150 годин / 5 кредитів ЄКТС
Семестровий контроль / контрольні заходи	Екзамен
Розклад занять	http://rozklad.kpi.ua/
Мова викладання	Українська
Інформація про керівника курсу / викладачів	<u>Лектор:</u> доцент, к.т.н. Шпурик Вадим Вадимович <u>Практичні заняття:</u> доцент, к.т.н. Шпурик Вадим Вадимович старший викладач Оленєва Ксенія Миколаївна <u>Лабораторні роботи:</u> доцент, к.т.н. Шпурик Вадим Вадимович, старший викладач Оленєва Ксенія Миколаївна
Розміщення курсу	Кампус – campus.kpi.ua http://route.ho.ua/kpi/c1/

Програма навчальної дисципліни

1. Опис навчальної дисципліни, її мета, предмет вивчення та результати навчання

Майбутньому фахівцеві варто вивчати дисципліну «Основи програмування» тому, що вона є базовим курсом для всіх тих, хто хоче вивчити мови програмування, зокрема, C і C++, та стати досвідченим розробником програмного забезпечення. Курс містить також спеціальні розділи, присвячені життєвому циклу програмних засобів сучасної ІТ-індустрії, проблеми помилок в програмах і методах верифікації програмного забезпечення, стилю програмування. Після проходження лабораторного практикуму студент буде здатний писати програми середньої складності на мовах C, C++. Процес побудований таким чином, що слухачі навчаються не тільки писати код і розуміти код, написаний іншими програмістами, але і вирішувати проблеми, з якими неминуче доведеться зіткнутися при створенні програмного забезпечення. Слід окремо підкреслити, що хоча цей курс викладається українською мовою, в цьому курсі ви паралельно вивчаєте і вдосконалюєте свою англійську мову і специфічну термінологію ІТ-галузі. Це допоможе як з розумінням програмування (більшість документації по мовам програмування написана саме англійською мовою), так і в подальшому з пошуком роботи.

Метою опанування дисципліни є освоєння фундаментальних основ програмування; ознайомлення з усіма основними термінами і поняттями; здобуття практичних навичок розробки алгоритмів; отримання реального досвіду розробки; навчитися писати доволі складні програми; познайомитися з передовими мовами програмування і напрямками програмування; чітко розуміти, який напрямок вам найцікавіше; отримати матеріали та інструкції для подальшого розвитку.

Предметом кредитного модуля є вивчення фундаментальних принципів сучасного процедурного та об'єктно-орієнтованого програмування на базі мови системного програмування С.

Програмні результати навчання

Дисципліна сприяє формуванню у студентів певних компетентностей, тобто здатностей до виконання певних завдань самостійно. Компетентності формуються як фахові, так і загальні.

Загальні компетентності:

- здатність до абстрактного мислення, аналізу та синтезу (ЗК 01);
- здатність до пошуку, оброблення та аналізу інформації з різних джерел (ЗК 06).

Фахові компетентності:

- здатність ідентифікувати, класифікувати та формулювати вимоги до програмного забезпечення (ФК 01);
- здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування (ФК 02);
- здатність розробляти архітектури, модулі та компоненти програмних систем (ФК 03);
- володіння знаннями про інформаційні моделі даних, здатність створювати програмне забезпечення для зберігання, видобування та опрацювання даних (ФК 07);
- здатність застосовувати фундаментальні і міждисциплінарні знання для успішного розв'язання завдань інженерії програмного забезпечення (ФК 08);
- здатність накопичувати, обробляти та систематизувати професійні знання щодо створення і супроводження програмного забезпечення та визнання важливості навчання протягом всього життя (ФК 10);
- здатність реалізовувати фази та ітерації життєвого циклу програмних систем та інформаційних технологій на основі відповідних моделей і підходів розробки програмного забезпечення (ФК 11);
- здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення (ФК 13);
- здатність до алгоритмічного та логічного мислення (ФК 14);
- володіти скриптовими та декларативними мовами програмування (ФК 16).

Згідно з вимогами програми навчальної дисципліни після засвоєння навчальної дисципліни студенти мають продемонструвати певні **програмні результати навчання**, тобто набуті ними знання й уміння описані далі:

- аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки (ПРН 01);
- знати основні процеси, фази та ітерації життєвого циклу програмного забезпечення (ПРН 03);
- уміти вибирати та використовувати відповідну задачі методологію створення програмного забезпечення (ПРН 06);

- знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення (ПРН 07);
- знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань (ПРН 13);
- мотивовано обирати мови програмування та технології розробки для розв'язання завдань створення і супроводження програмного забезпечення (ПРН 15);
- знати та вміти застосовувати інформаційні технології обробки, зберігання та передачі даних (ПРН 18).

2. Пререквізити та постреквізити дисципліни

Вивчення дисципліни «Основи програмування» проходить паралельно з вивченням студентами основоположних дисциплін за спеціальністю, тому розміщується дисципліна у структурно-логічній схемі паралельно з нормативними освітніми компонентами циклу загальної підготовки та з дисциплінами циклу професійної підготовки, такими як: «Алгоритми та структури даних», «Основи комп'ютерних систем і мереж», «Компоненти програмної інженерії. Частина 1. Вступ до програмної інженерії» та «Комп'ютерна дискретна математика».

Дисципліна «Основи програмування» забезпечує вивчення таких дисциплін, як: «Бази даних», «Компоненти програмної інженерії», «Об'єктно-орієнтований аналіз та конструювання програмних систем», а також забезпечує написання курсової роботи з циклу професійної підготовки «Основи програмування. Курсова робота».

3. Зміст навчальної дисципліни

Навчальна дисципліна «Основи програмування» складається з двох кредитних модулів, перший з них – «Основи програмування. Частина 1. Базові конструкції» має структуру представлену в таблиці далі.

Розділ 1. Введення в мову C: основні поняття	Тема – 1. Введення в мову C
Розділ 2. Дані та конструкції мови C	Тема – 2.1. Дані мови
	Тема – 2.2. Керуючі конструкції. Оператори управління
	Тема – 2.3. Введення та виведення даних
Розділ 3. Конструкції та складові мови C	Тема – 3.1. Функції
	Тема – 3.2. Вказівники: основні відомості, принцип роботи, використання
	Тема – 3.3. Вказівники та масиви
	Тема – 3.4. Функції та масиви
	Тема – 3.5. Символьні рядки та рядкові функції
Розділ 4. Організація програми, взаємодія з файлами мовою C	Тема – 4.1. Організація програми
	Тема – 4.2. Файлова взаємодія
Розділ 5. Додаткові відомості мови C	Тема – 5.1. Додаткові можливості використання вказівників
	Тема – 5.2. Структури даних

4. Навчальні матеріали та ресурси

Для кращого засвоєння дисципліни, виконання лабораторних і самостійної роботи (СРС) студентам надається список рекомендованої літератури та навчальні матеріали, розміщені в Інтернеті та на університетських ресурсах.

Ресурси та література:

1. Конспект лекцій. 1 семестр – URL: <http://route.ho.ua/kpi/c1/>
2. Основи програмування. Професійний підхід [Електронний ресурс] : навчальний посібник для здобувачів ступеня бакалавра за освітньо-професійною програмою «Інженерія

програмного забезпечення інтелектуальних кібер-фізичних систем в енергетиці» спеціальності 121 Інженерія програмного забезпечення / В. В. Шпурик, К. М. Оленєва ; КПІ ім. Ігоря Сікорського. – Київ : КПІ ім. Ігоря Сікорського, 2024. – 106 с. – URL: <https://ela.kpi.ua/items/78415162-2331-4510-a9d6-f8cf20c3ed7f>

3. Брайан У. Керніган, Денніс М. Рітчі. Мова програмування C, 2-ге видання. – URL: <https://www.amazon.com/Programming-Language-2nd-Brian-Kernighan>

4. Herbert Schildt. C: The Complete Reference (Complete Reference Series). – URL: <https://www.amazon.com/C-Complete-Reference>

5. Stephen G. Kochan. Programming in C 3rd Edition. – URL: <http://www.williamspublishing.com/Books/5-8459-1088-9.html>

Навчальний контент

1. Методика опанування навчальної дисципліни (освітнього компонента)

Навчальна дисципліна складається з двох кредитних модулів, які охоплюють 11 кредитів або 330 годин. За порядком слідування та розподілом годин кредитні модулі розміщуються на першому курсі в осінньому та весінньому семестрах.

В осінньому семестрі першого курсу передбачено перший кредитний модуль дисципліни «Основи програмування» – освітній компонент «Основи програмування. Частина 1. Базові конструкції», на який відводиться 5 кредитів або 150 годин.

Перший кредитний модуль дисципліни включає в себе проведення протягом семестру:

- лекційних занять – 30 годин;
- практичних занять (комп'ютерні практикуми) – 14 годин;
- лабораторних робіт – 30 годин.

Також в освітньому компоненті відводиться 76 годин на самостійну роботу студента.

Календарний план запланованих заходів: лекційні заняття

№	Розподіл лекцій за темами	
1	Тема – 1. Введення в мову C	Лекція 1. Вступ до мови C. Знайомство з компілятором GCC <i>перелік тем: виникнення мови, компілятор, кроки програмування, gcc, структура програми, декларування та виклик програми, синтаксичні та семантичні помилки; основні типи даних</i>
2	Тема – 2.1. Дані мови	Лекція 2. Дані мови: символічні рядки, оператори, вирази та конструкції <i>перелік тем: використання текстової інформації у вигляді рядка, різниця між string і char типами, макро- та інші константи, операції ін- та декрементації</i>
3	Тема – 2.2. Керуючі конструкції. Оператори управління	Лекція 3. Оператори управління: поняття «цикли» <i>перелік тем: види циклів та принципи роботи з ними, приклади коли та навіщо потрібні цикли</i>
4		Лекція 4. Оператори управління: розгалуження та переходи <i>перелік тем: вкладені цикли, приклади та розбір взаємодії різних циклів одне з одним</i>
5	Тема – 2.3. Введення та виведення даних	Лекція 5. Символьне введення та виведення даних <i>перелік тем: символічний ввід/вивід змінних, бібліотека <code>sture.h</code>, EOF, проблеми вводу/виводу, змішування типів даних на виводі, валідація вводу, створення програмного меню</i>
6	Тема – 3.1. Функції	Лекція 6. Функції. Частина перша: створення та використання простих функцій

		<i>перелік тем: поняття функцій, створення простих функцій, принцип і нюанси роботи з return, рекурсія, розв'язання задачі «Ханойські вежі»</i>
7		Лекція 7. Функції. Частина друга: створення та використання власних функцій <i>перелік тем: створення власних функцій для вирішення задач, використання функцій при роботі з масивами, різні алгоритми сортування, enum і typedef</i>
8	Тема – 3.2. Вказівники: основні відомості, принцип роботи, використання	Лекція 8. Вказівники. Частина перша: основні поняття та відомості <i>перелік тем: що таке вказівники, принцип їх роботи, як використовувати посилання та адреси, приклади в яких вказівники полегшують роботу або коли без вказівників не обійтись</i>
9	Тема – 3.3. Вказівники та масиви	Лекція 9. Вказівники. Частина друга: використання вказівників для роботи з масивами <i>перелік тем: segmentation fault, детальніше про керування вказівниками та адресами, використання вказівників для роботи з багатовимірними масивами, контроль границь масиву, використання вказівників при роботі з рядковими даними</i>
10		Лекція 10. Вказівники. Частина третя: використання вказівників у різних типах сортувань <i>перелік тем: більше про роботу вказівників з масивами, взаємозв'язок між адресами, вмістом масиву та вказівниками, вказівник на вказівник, можливості використання вказівників, вказівники в алгоритмах сортувань</i>
11	Тема – 3.4. Функції та масиви	Лекція 11. Функції та масиви <i>перелік тем: альтернативи синтаксису прототипів, вказівник на N-вимірний масив, складений літерал, вказівник на масив рядків, вказівники на функції, приклад використання псевдокоду</i>
12	Тема – 3.5. Символьні рядки та рядкові функції	Лекція 12. Символьні рядки та рядкові функції <i>перелік тем: операції з рядковими літералами, функція fgets(), доступ до символів у рядку, функції strlen(), strcat() і strncat(), strcmp(), strcpy() і strncpy(), сортування рядків</i>
13	Тема – 4.1. Організація програми	Лекція 13. Організація програми. Інтерфейс <i>перелік тем: організація програми, н'яті класів зберігання (сховища), область видимості; локальні, зовнішні, автоматичні та статичні змінні; приклади використання та поширені помилки</i>
14	Тема – 4.2. Файлова взаємодія	Лекція 14. Взаємодія з файлами, файлове введення та виведення <i>перелік тем: аргументи командного рядка, перетворення рядків у числа; робота з файлами, доступ до і пошук по директорії, відкриття/закриття файлів, пошук у файлі, копіювання з файлу у файл; бінарний ввід/вивід: функції fread() і fwrite()</i>
15	Тема – 5.1. Додаткові можливості використання вказівників	Лекція 15. Розширене використання вказівників <i>перелік тем: динамічне виділення пам'яті; Big O нотація: функції malloc(), free(); виділення пам'яті, витік пам'яті; загальні функції середовища виконання</i>

Календарний план запланованих заходів: практичні заняття

№	Назва роботи
1	Практичне заняття 1. Знайомство з компілятором GCC: виконання та компіляція першої програми. Дані мови C: символьні рядки, оператори, вирази та твердження
2	Практичне заняття 2. Робота з вкладеними циклами. Символьне введення та виведення даних
3	Практичне заняття 3. Функції. Частина перша та друга: використання простих і власних функцій
4	Практичне заняття 4. Вказівники. Частина друга: використання вказівників для роботи з масивами
5	Практичне заняття 5. Символьні рядки та рядкові функції. Частина перша: робота з рядковими даними
6	Практичне заняття 6. Організація програми. Файлове введення та виведення даних
7	Практичне заняття 7. Структури даних, об'єднання та перелік

Календарний план запланованих заходів: лабораторні роботи

№	Назва роботи
1	Лабораторна робота 1. Робота з циклами. Робота з вкладеними циклами
2	Лабораторна робота 2. Розгалуження та переходи в циклах
3	Лабораторна робота 3. Символьне введення/виведення даних
4	Лабораторна робота 4. Функції. Частина перша: використання простих функцій
5	Лабораторна робота 5. Функції. Частина друга: використання власних функцій
6	Лабораторна робота 6.1. Вказівники. Частина перша та друга: основні поняття та відомості, використання вказівників для роботи з масивами
7	Лабораторна робота 6.2. Вказівники. Частина перша та друга. Продовження
8	Лабораторна робота 7. Вказівники. Частина третя: використання вказівників у різних типах сортувань
9	Лабораторна робота 8.1. Функції та масиви
10	Лабораторна робота 8.2. Функції та масиви. Продовження
11	Лабораторна робота 9. Символьні рядки та рядкові функції. Частина друга: створення власних функцій для роботи з рядковими даними
12	Лабораторна робота 10. Організація програми. Інтерфейс
13	Лабораторна робота 11. Взаємодія з файлами, файлове введення/виведення даних
14	Лабораторна робота 12. Структури даних
15	Лабораторна робота. Проведення модульної контрольної роботи

2. Самостійна робота студента

Самостійна робота студента в кредитному модулі «Основи програмування. Частина 1. Базові конструкції» розрахована на 76 годин і поділена на завдання відповідно до розділів і тем.

Самостійна робота складається з завдань, які студенти мають виконати самостійно в поза аудиторні години навчання. Завдання містять як теоретичні, так і практичні задачі та розташовані в календарному плані дисципліни так, щоб доповнювати лекційний матеріал або передувати практичним заняттям чи виконанню лабораторних робіт.

Також до годин самостійної роботи студентів входять підготовка до модульної контрольної роботи та екзамену.

Далі наведений перелік завдань.

Календарний план запланованої самостійної роботи студентів

№	Назви самостійних робіт за розділами та темами		Кіл-сть годин СРС
Розділ 2. Дані та конструкції мови C			
1	Тема – 1. Дані мови	СРС 1. Можливості використання уявних типів даних	2
2	Тема – 2. Керуючі конструкції.	СРС 2. Випадки використання циклів з перед- та післяумовами (do і do-while)	2
3	Оператори управління	СРС 3. Оптимізація вкладених циклів з метою підвищення ефективності коду	2
4		СРС 4. Використання операторів break і continue при роботі з циклами	2
5	Тема – 3. Введення та виведення даних	СРС 5. Різниця між getchar(), gets() і scanf(), можливості їх комбінування з printf()	2
Розділ 3. Конструкції та складові мови C			
6	Тема – 1. Функції	СРС 6. Робота функції return: основні поняття, виключення та помилки	2
7		СРС 7. Випадки, коли використання рекурсії є обов’язковим	2
8		СРС 8. Робота з даними типу enum	2
9	Тема – 2. Вказівники: основні відомості, принцип роботи, використання	СРС 9. Одно-, дво- та багатовимірні масиви: основні відомості	1
10		СРС 10. Аналіз проблем і повідомлень компілятора, що виникають при використанні неініційованих вказівників	2
11	Тема – 3. Вказівники та масиви	СРС 11. Вимірювання часу, який витрачається програмою при використанні різних типів сортувань	2
12	Тема – 4. Функції та масиви	СРС 12. Які бувають помилки та причини їх виникнення при використанні складеного літералу	2
13	Тема – 5. Символьні рядки та рядкові функції	СРС 13. Використання UTF-8 зокрема символів кириличної абетки у програмі	3
Розділ 4. Організація програми, взаємодія з файлами мовою C			
14	Тема – 1. Організація програми	СРС 14. Ознаки «хорошого» інтерфейсу	3
15	Тема – 2. Файлова взаємодія	СРС 15. Порівняння двійкового та текстового режимів	3
Розділ 5. Додаткові відомості мови C			
16	Тема – 1. Додаткові можливості	СРС 16. Використання конструкцій «вказівник на вказівник». Коректне розіменування вказівників	4

	використання вказівників		
17	Тема – 2. Структури даних	СРС 17. Масиви структур: використання та застосування	4
Контрольні заходи			
18	Підготовка до модульної контрольної роботи		6
19	Підготовка до екзамену		30

Політика та контроль

1. Політика навчальної дисципліни (освітнього компонента)

В освітньому компоненті «Основи програмування. Частина 1. Базові конструкції» протягом семестру передбачено проведення: лекційних і практичних занять, лабораторних робіт і модульної контрольної роботи (МКР).

Відвідування занять

Відвідування усіх видів аудиторних занять, а саме: лекційних, практичних і лабораторних є обов'язковим.

Виняток становить лише відсутність на заняттях з поважних причин (тобто хвороби, форс-мажорні обставини, участь в олімпіадах, тощо), про що студент має повідомити викладача завчасно та надати підтвердження причини. У разі відсутності студента на аудиторному занятті з поважних причин нарахування штрафних балів не здійснюється.

Правила захисту лабораторних робіт

Протягом семестру студенти мають виконувати і захищати лабораторні роботи. Для кожної з робіт встановлені граничні терміни захисту (дедлайни).

Для отримання максимальних балів за кожну з передбачених програмою робіт студент повинен виконати роботу самостійно, в повній мірі, без помилок, вчасно здати звіт з роботи та вчасно роботу захистити (захист очний).

Самостійне виконання роботи має на увазі виконання роботи студентом на основі лекційних і практичних матеріалів, за необхідності використання додаткових рекомендованих джерел. Самостійне виконання забороняє використання для розв'язку завдань матеріалів інших студентів (чи випускників кафедри), матеріали (програмні коди) згенеровані штучним інтелектом чи матеріали скопійовані з будь-яких доступних ресурсів через Інтернет. Також лабораторні роботи мають виконуватись з урахуванням вивченого матеріалу в рамках дисципліни, а не чогось іншого, вивченого з інших джерел, не затверджених для дисципліни.

Якщо студент пропустив термін захисту роботи з поважної причини, він має можливість виконати та здати роботу без втрати балів.

Якщо студент пропустив термін захисту роботи без поважних причин – йому надається можливість відпрацювання пропущеної роботи шляхом написання самостійної роботи за відповідною темою. Але в такому випадку максимальний бал за роботу становитиме 60% від найвищої оцінки за завдання.

Причиною, що унеможливорює захист пропущених лабораторних робіт є дедлайн на відповідну роботу та закінчення навчального процесу (семестру).

Політика дедлайнів та перескладань

На кожну лабораторну роботу за календарним планом кредитного модуля відводиться по два тижні для виконання та захисту роботи. Всі терміни оголошуються студентам на початку семестру.

Дедлайни можуть коригуватись протягом семестру у випадку форс-мажорних ситуацій. Перескладання лабораторних робіт не передбачено.

Відпрацювання пропущеної роботи передбачено у вигляді самостійної роботи протягом двох тижнів після завершення основного дедлайну на захист відповідної роботи.

Перескладання модульної контрольної роботи не передбачено. Написання модульної контрольної роботи поза визначеною датою можливе лише за поважних причин, що мають підтвердження.

2. Види контролю результатів навчання

Для кредитного модуля «Основи програмування. Частина 1. Базові конструкції» передбачені наступні види контролю:

- проведення поточного контролю;
- проведення календарного контролю;
- проведення семестрового контролю.

Поточний контроль

Поточний контроль протягом семестру проводиться у вигляді двох самостійних робіт і модульної контрольної роботи. Поточний контроль має на меті перевірку знань та вмінь студентів, засвоєних і здобутих ними на усіх видах аудиторних занять.

Модульна контрольна робота проводиться за матеріалами усіх аудиторних видів занять, тобто: лабораторних робіт, лекційних і практичних занять. Завдання складаються з теоретичних і практичних запитань. Контрольна робота проводиться в кінці семестру. Питання та опорні теми для повторення оголошуються студентам заздалегідь.

Для проведення модульної контрольної роботи виділяються одна учбова година за рахунок лабораторного заняття.

Самостійні роботи проводяться за матеріалами лекційних і практичних занять, лабораторних робіт, складаються з теоретичних і практичних завдань. Самостійні роботи передбачені для перевірки поточних знань студентів, для проведення календарного контролю та не входять у загальний рейтинг студента. Опорні теми до повторення оголошуються студентам заздалегідь.

Календарний контроль

Календарний контроль проводиться двічі на семестр як моніторинг поточного стану знань та вмінь студентів і виконання ними умов навчальної програми освітнього компонента.

Умовою отримання атестації (позитивної оцінки) за проходження календарного контролю є виконання вимог за кількістю та якістю виконаних лабораторних робіт, а також отримання позитивної оцінки (≥60% від максимального балу) за написання відповідної самостійної роботи.

Причинами, що призводять до отримання незадовільного результату атестації під час календарного контролю є отримання незадовільної оцінки за самостійну роботу та захист передбачених календарним планом лабораторних робіт на незадовільні бали або ж у кількості менше 60% запланованих.

Семестровий контроль

Семестровий контроль з кредитного модуля здійснюється у вигляді екзамену.

Умовами допуску до здачі екзамену є виконання завдань, передбачених лабораторними роботами у кількості ≥ 75% від загальної кількості лабораторних. А також, написання модульної контрольної роботи на оцінку не нижче оцінки «достатньо» (60% від максимального балу).

Загальний рейтинг студента за семестр для допуску до екзамену має становити не менше 36 балів.

Екзамен проводиться в письмовому вигляді, містить теоретичні питання та практичні завдання. Перелік питань та опорні теми для підготовки до екзамену заздалегідь надаються студентам.

3. Рейтингова система оцінювання результатів навчання

Система оцінювання результатів навчання

Оцінювання результатів навчання здійснюється згідно рейтингової системи оцінювання за 100-бальною шкалою.

Рейтинг студента складається з балів за захист лабораторних робіт, написання модульної контрольної роботи та складання іспиту. Протягом семестру студент може набрати 48 балів за лабораторні роботи та 12 балів за МКР, що в сумі дає 60 балів.

Також під час роботи на лекційних і практичних заняттях студенти можуть отримати заохочувальні (додаткові) бали. За невиконання умов навчання передбачені штрафні бали. Кількість заохочувальних або штрафних балів рівна 5 балам на семестр.

Оцінювання лабораторних робіт

Для освітнього компонента «Основи програмування. Частина 1. Базові конструкції» передбачено 12 лабораторних робіт на оцінку. Загалом за лабораторні роботи за семестр студент може набрати 48 балів. Для отримання оцінки за лабораторну роботу потрібно виконати роботу, здати звіт і захистити роботу. Без попередньо зданого звіту до захисту студенти не допускаються.

Оцінюються роботи по 4 бали кожна за наступними критеріями:

- бездоганна робота оцінюється в 4 бали – завдання виконано в повній мірі, вирішує всі поставлені задачі, виконано без помилок, вчасно здано звіт у належній якості та виконано очний захист на відмінно;

- є певні недоліки у виконанні роботи – 3 бали – завдання виконано в повній мірі, але допущені незначні помилки, вчасно здано звіт у належній якості та виконано очний захист на дуже добре/добре;

- є певні недоліки в виконанні роботи – 2-1 бали – завдання виконано неповністю, допущені суттєві помилки, вчасно здано звіт в неналежній якості чи допущені незначні помилки, очний захист виконано на задовільно/достатньо;

- робота не виконана, не захищена або не зданий звіт – 0 балів – робота здана лише у вигляді звіту, але не захищена особисто; робота здана особисто, але немає звіту; робота виконана чи захищена менш, ніж на 60% або здана пізніше дедлайну.

Оцінювання самостійних робіт

Самостійні роботи призначені для проведення календарного контролю оцінюються в 10 балів. Критерії оцінювання самостійних робіт аналогічні критеріям, що встановлені до оцінювання модульної контрольної роботи (описані далі).

Самостійні роботи, що виконуються студентами з умовою відпрацювання пропущених робіт без поважної причини оцінюються в 60% від максимального балу (тобто в 2,4 бали) за критеріями відповідними лабораторним роботам, без здачі звіту.

Оцінювання модульної контрольної роботи

Контрольна робота складається із чотирьох завдань: двох теоретичних питань і двох практичних завдань. Оцінювання завдань контрольної роботи виконується за таким розподілом: теоретичні питання оцінюються по 1,5 бали, а практичні по 4 і 5 балів.

Оцінюються питання модульної контрольної роботи за такими критеріями:

- «відмінно», повна відповідь, не менше 90% потрібної інформації, програмний код написаний згідно стандартів кодування, вирішує основне завдання, включає всі необхідні програмні модулі, конструкції та хоча б частину перевірок; відповіді написані точними формулюваннями та розгорнуто;

- «добре», достатньо повна відповідь, не менше 75% потрібної інформації, що виконана згідно з вимогами, є незначні неточності, відсутні перевірки та код неструктурований; відповіді короткі, але точні;

– «задовільно», неповна відповідь, не менше 60% потрібної інформації, завдання виконане з певними недоліками, в коді відсутні перевірки, наявні помилки; відповіді неповні чи містять суттєві помилки;

– «незадовільно», відповідь не відповідає умовам до «достатньо», наявні суттєві помилки, відповідь відсутня або не вирішує поставлені задачі; наявне копіювання, списування роботи, використання сторонньої допомоги (людини, штучного інтелекту, тощо).

Оцінювання екзаменаційної роботи

Умовою допуску до екзамену є виконання завдань, передбачених лабораторними роботами у кількості ■ 75% від загальної кількості лабораторних, написання модульної контрольної роботи на ■ 7,2 бали і стартовий рейтинг не менше 36 балів.

Екзамен студенти виконують письмово. Білети складаються з теоретичних питань і практичних завдань. Теоретичні запитання оцінюються по 7 балів кожне, практичні завдання по 13 балів кожне, за наступними критеріями:

– «відмінно», повна відповідь, не менше 90% потрібної інформації, що виконана згідно з вимогами до рівня вмінь;

– «добре», достатньо повна відповідь, не менше 75% потрібної інформації, що виконана згідно з вимогами до рівня вмінь або є незначні неточності;

– «задовільно» і «достатньо», неповна відповідь, не менше 60% потрібної інформації, завдання виконане з суттєвими недоліками;

– «незадовільно», відповідь не відповідає умовам до «достатньо» – 0 балів.

Сума стартових балів та балів за екзаменаційну роботу переводиться до екзаменаційної оцінки згідно з таблицею:

Бали	Оцінка
100...95	Відмінно
94...85	Дуже добре
84...75	Добре
74...65	Задовільно
64...60	Достатньо
Менше 60	Незадовільно
Не виконані умови допуску	Не допущено

За умови проведення навчання у дистанційному режимі екзамен не проводиться. В такому випадку стартові бали, отримані за виконання лабораторних робіт і написання МКР, переводяться у 100-бальну шкалу шляхом приведення через пропорцію.

Робочу програму навчальної дисципліни (силабус):

Складено

доцент, к.т.н., Шпурик Вадим Вадимович,
старший викладач Оленєва Ксенія Миколаївна

Ухвалено

кафедрою ІПЗЕ (протокол № 42 від 18.06.2025 р.)

Погоджено

методичною комісією ННІАТЕ (протокол № 8 від 20.06.2025 р.)