



ОСНОВИ ПРОГРАМУВАННЯ

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти	<i>Перший (бакалаврський)</i>
Галузь знань	<i>G Інженерія, виробництво та будівництво</i>
Спеціальність	<i>G3 Електрична інженерія</i>
Освітня програма	<i>Енергоменеджмент та енергоефективні технології</i>
Статус дисципліни	<i>Нормативна</i>
Форма навчання	<i>Очна (денна)</i>
Рік підготовки, семестр	<i>1 курс, осінній семестр</i>
Обсяг дисципліни	<i>180 годин / 6 кредитів (лекції – 30 год., практичні заняття – 46 год., СРС – 104 год.)</i>
Семестровий контроль/ контрольні заходи	<i>Екзамен / модульна контрольна робота</i>
Розклад занять	https://schedule.kpi.ua/
Мова викладання	<i>Українська</i>
Інформація про керівника курсу / викладачів	Лектор: к.т.н, доцент, Філянін Данило Володимирович, filyanin.daniel@edu.kpi.ua Практичні заняття: к.т.н, доцент, Філянін Данило Володимирович Консультації: щопонеділка, 16:00-17:00
Розміщення курсу	https://classroom.google.com/c/MTQ1NjEyODQ5NDQ1?cjc=4c3uw5t

Програма навчальної дисципліни

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

Поява персональних комп'ютерів (ПК) значно розширило коло користувачів обчислювальної техніки, яке містить тепер не тільки професійних програмістів, але й фахівців різних областей діяльності. Змінився і склад програмного забезпечення, в якому велике місце займають готові прикладні системи, що полегшують користувачеві застосування ПК для різних видів робіт. Проте нерідко виникає потреба модифікувати або розширити деякі функції готового програмного забезпечення. Саму прикладну систему теж треба програмувати на якійсь мові.

Метою навчальної дисципліни є формування у здобувачів вищої освіти відповідного рівня знань і досвіду використання сучасних методів програмування для вирішення інженерних завдань у сфері електро- та енергопостачання; розробки програмних проектів для комп'ютерів на основі використання технології об'єктно-орієнтованого програмування; створення та застосовування алгоритмів для вирішення типових задач обробки інформації.

Предметом навчальної дисципліни є методи та алгоритми розв'язання інженерних задач за допомогою сучасних алгоритмічних мов з використанням сучасних комп'ютерних технологій.

Компетентності: (ЗК01) Здатність до абстрактного мислення, аналізу і синтезу; (ЗК02) здатність застосовувати знання у практичних ситуаціях; (ЗК03) здатність спілкуватися державною мовою як усно, так і письмово; (ЗК08) здатність працювати автономно; (ФК01) здатність вирішувати практичні задачі із застосуванням систем автоматизованого проектування і розрахунків (САПР).

Програмні результати навчання: (ПРН06) застосовувати прикладне програмне забезпечення, мікроконтролери та мікропроцесорну техніку для вирішення практичних проблем у професійній діяльності; (ПРН18) вміти самостійно вчитися, опановувати нові знання і вдосконалювати навички роботи з сучасним обладнанням, вимірювальною технікою та прикладним програмним забезпеченням.

2. Пререквізити та постреквізити дисципліни (місце в структурно-логічній схемі навчання за відповідною освітньою програмою)

Навчальна дисципліна «ОСНОВИ ПРОГРАМУВАННЯ» викладається на основі знань та умінь, одержаних здобувачами вищої освіти під час вивчення інформатики, математики, фізики на рівні шкільного курсу. Компетентності та програмні результати навчання, одержані в процесі вивчення навчальної дисципліни «ОСНОВИ ПРОГРАМУВАННЯ» є необхідними для вивчення дисципліни «Інформаційні системи і технології в електроенергетиці», «Математичні задачі енергетики» та подальшого якісного виконання бакалаврського дипломного проекту.

3. Зміст навчальної дисципліни

Розділ 1. Основи алгоритмізації та програмування

Тема 1.1. Принцип дії ЕОМ. Процесор. Архітектура процесора. Архітектура комп'ютера. Операційна система.

Тема 1.2. Арифметичні основи ЕОМ. Системи числення. Подання інформації в ЕОМ. Поняття алгоритму. Основи програмування.

Тема 1.3. Мови програмування.

Розділ 2. Об'єкти даних і базові засоби їх обробки

Тема 2.1. Основні елементи C#. Операції консольного введення-виводу.

Тема 2.2. Базові операції консольного введення-виводу. Типи даних.

Тема 2.3. Вирази та операції. Арифметичні операції. Порозрядні операції. Логічні операції.

Тема 2.4. Клас Math для математичних обчислень у C#.

Тема 2.5. Робота з текстовою інформацією. Символи. Рядки.

Розділ 3. Основні конструкції мови програмування C#

Тема 3.1. Поняття про структурне програмування. Структура послідовності C#. Структури вибору C#.

Тема 3.2. Структури повторення C#. Цикл за лічильником.

Тема 3.3. Цикли за умовою.

Тема 3.4. Масиви. Прості масиви.

Тема 3.5. Багатомірні масиви.

Тема 3.6. Клас Array. Сортування та пошук у масивах.

Тема 3.7. Списки. Переліки enum.

Тема 3.8. Процедури та функції. Методи.

4. Навчальні матеріали та ресурси

Базова література:

1. Обчислювальна техніка та програмування: Алгоритми та їх реалізація. Конспект лекцій [Електронний ресурс]: навч. посіб. для студ. спеціальності 141 Електроенергетика, електротехніка та електромеханіка / КПІ ім. Ігоря Сікорського; уклад.: Д. В. Філянін. – Електронні текстові дані (1 файл: 1,57 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2022. – 99 с. URL: <https://ela.kpi.ua/handle/123456789/47967> (дата звернення 10.06.2026).

2. Andrew Troelsen, Philip Japikse, "Pro C# 9 with .NET 5: Foundational Principles and Practices in Programming", Apress, P. 1411. April 2021.
3. Joseph Albahari, Ben Albahari, "C# 9.0 Pocket Reference: Instant Help for C# 9.0 Programmers", 1st edition, O'Reilly Media, P. 264. February 2021.
4. Mark J. Price, "C# 10 and .NET 6 – Modern Cross-Platform Development", Sixth Edition, Packt Publishing Ltd., P. 825. 2023.
5. ОБЧИСЛЮВАЛЬНА ТЕХНІКА ТА ПРОГРАМУВАННЯ: Алгоритми та їх реалізація. Комп'ютерний практикум [Електронний ресурс] : навч. посіб. для студ. спеціальності 141 Електроенергетика, електротехніка та електромеханіка та 184 Гірництво / КПІ ім. Ігоря Сікорського; уклад.: Д. В. Філянін, В. П. Опришко, В. О. Броницький, О. Е. Максименко. – Електронні текстові дані (1 файл: 1,21 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2022. – 95 с. URL: <https://ela.kpi.ua/handle/123456789/47964> (дата звернення 10.06.2026).

Допоміжна література:

6. Mikael Olsson, "C# 10 Quick Syntax Reference. 4th Ed.", Apress, P. 196. 2022.
7. Мова програмування C#. URL: <http://www.znannya.org/?view=csharp> (дата звернення 10.06.2026).
8. C# language documentation. URL: <https://docs.microsoft.com/en-us/dotnet/csharp/> (дата звернення 10.06.2026).
9. Visual C# Guided Tour. URL: [https://docs.microsoft.com/en-us/previous-versions/visualstudio/visual-studio-2008/bb383962\(v=vs.90\)?redirectedfrom=MSDN](https://docs.microsoft.com/en-us/previous-versions/visualstudio/visual-studio-2008/bb383962(v=vs.90)?redirectedfrom=MSDN) (дата звернення 10.06.2026).
10. C# Programming. URL: https://en.wikibooks.org/wiki/C_Sharp_Programming (дата звернення 10.06.2026).
11. C# / CSharp Tutorial. URL: <http://www.java2s.com/Tutorial/CSharp/CatalogCSharp.htm> (дата звернення 10.06.2026).
12. C#. URL: https://www.bestprog.net/uk/sitemap_ua/c-3/ (дата звернення 10.06.2026).
13. Visual Studio 2022 Community. Безкоштовне повнофункціональне середовище IDE, що розширюється, для створення сучасних програм Android, iOS і Windows, а також веб-додатків і хмарних служб.

Літературу, бібліографія якої подана із посиланням, можна знайти в інтернеті. Літературу, бібліографія якої не містить посилання, можна знайти в бібліотеці КПІ ім. Ігоря Сікорського.

Обов'язковим для прочитання є базова література [1]-[5]. Базова література, що є обов'язковою для прочитання, а також зв'язок цих ресурсів з конкретними темами дисципліни наводиться нижче, в методиці опанування навчальної дисципліни. Усі інші літературні джерела є факультативними, з ними рекомендується ознайомитись.

Навчальний контент

5. Методика опанування навчальної дисципліни (освітнього компонента)

Навчальна дисципліна охоплює 30 годин лекцій та 46 годин практичних занять та виконання модульної контрольної роботи (МКР) тривалістю 2 акад. год.

Виходячи з розподілу часу на вивчення дисципліни, рекомендується чотирнадцять лекцій та одне лекційне зайняття виділяється під виконання МКР. При проведенні лекційних занять використовується приміщення з мультимедійним проекторним обладнанням. При проведенні практичних занять використовуються такі сучасні програмні засоби, як програмне середовище Visual Studio 2026 та середовище Google Classroom, де розташовуються основні лекційні та практичні матеріали.

Лекційні заняття

№ з/п	Назва теми лекції та перелік основних питань (посилання на літературу та завдання на СРС)
1	ТЕМА 1. Принцип дії ЕОМ. Процесор. Архітектура процесора. Архітектура комп'ютера. Операційна система. Література: [1, 2]. Завдання на СРС: Використовуючи рекомендовану літературу, підготувати відповіді на теоретичні запитання до тем, що розглядались на лекції.
2	ТЕМА 2. Арифметичні основи ЕОМ. Системи числення. Подання інформації в ЕОМ. Поняття алгоритму. Література: [1, 2]. Завдання на СРС: Використовуючи рекомендовану літературу, підготувати відповіді на теоретичні запитання до тем, що розглядались на лекції.
3	ТЕМА 3. Мови програмування. Базові елементи C#. Література: [1, 2]. Завдання на СРС: Використовуючи рекомендовану літературу, підготувати відповіді на теоретичні запитання до тем, що розглядались на лекції.
4	ТЕМА 4. Базові операції консольного введення-виводу. Типи даних. Література: [1, 2]. Завдання на СРС: Використовуючи рекомендовану літературу, підготувати відповіді на теоретичні запитання до тем, що розглядались на лекції.
5	ТЕМА 5. Вирази та операції. Арифметичні операції. Порозрядні операції. Логічні операції. Клас Math для математичних обчислень у C #. Література: [1, 2]. Завдання на СРС: Використовуючи рекомендовану літературу, підготувати відповіді на теоретичні запитання до тем, що розглядались на лекції.
6	ТЕМА 6. Робота з текстовою інформацією. Символи. Рядки. Література: [1, 2]. Завдання на СРС: Використовуючи рекомендовану літературу, підготувати відповіді на теоретичні запитання до тем, що розглядались на лекції.
7	ТЕМА 7. Поняття про структурне програмування. Структура послідовності C#. Структури вибору C#. Література: [1, 2]. Завдання на СРС: Використовуючи рекомендовану літературу, підготувати відповіді на теоретичні запитання до тем, що розглядались на лекції.
8	ТЕМА 8. . Структури повторення C#. Цикл за лічильником. Література: [1, 2]. Завдання на СРС: Використовуючи рекомендовану літературу, підготувати відповіді на теоретичні запитання до тем, що розглядались на лекції.
9	ТЕМА 9. Цикли за умовою. Література: [1, 2]. Завдання на СРС: Використовуючи рекомендовану літературу, підготувати відповіді на теоретичні запитання до тем, що розглядались на лекції.
10	ТЕМА 10. Масиви. Прості масиви. Література: [1, 2]. Завдання на СРС: Використовуючи рекомендовану літературу, підготувати відповіді на теоретичні запитання до тем, що розглядались на лекції.

11	ТЕМА 11. Багатомірні масиви. Література: [1, 2]. Завдання на СРС: Використовуючи рекомендовану літературу, підготувати відповіді на теоретичні запитання до тем, що розглядались на лекції.
12	ТЕМА 12. Клас Array. Сортуння та пошук у масивах. Література: [1, 2]. Завдання на СРС: Використовуючи рекомендовану літературу, підготувати відповіді на теоретичні запитання до тем, що розглядались на лекції.
13	ТЕМА 13. Списки. Переліки епит. Література: [1, 2]. Завдання на СРС: Використовуючи рекомендовану літературу, підготувати відповіді на теоретичні запитання до тем, що розглядались на лекції.
14	ТЕМА 14. Процедури та функції. Методи. Література: [1, 2]. Завдання на СРС: Використовуючи рекомендовану літературу, підготувати відповіді на теоретичні запитання до тем, що розглядались на лекції.
15	Модульна контрольна робота.

Практичні заняття

Основним завданням циклу практичних занять є закріплення матеріалу, що було розглянуто на практичних заняттях, та набуття умінь їх застосування на обчислювальній техніці шляхом виконання відповідних індивідуальних завдань.

№ з/п	Назва теми заняття та перелік основних питань (перелік дидактичного забезпечення, посилання на літературу та завдання на СРС)
1	ТЕМА 1. Вступне заняття.
2	ТЕМА 2. Знайомство з середовищем Visual Studio. Розробка елементарної програми мовою програмування C#. Література: [1, 6]. Завдання на СРС: Запам'ятати мету та алгоритм вирішення завдань, що розглядалися на практичному занятті, підготуватися до самостійного вирішення подібних завдань.
3	ТЕМА 3. Розробка елементарної програми мовою програмування C#. Література: [1, 6]. Завдання на СРС: Запам'ятати мету та алгоритм вирішення завдань, що розглядалися на практичному занятті, підготуватися до самостійного вирішення подібних завдань.
4	ТЕМА 4. Базові типи даних. Форматування консольного виведення. Література: [1, 6]. Завдання на СРС: Запам'ятати мету та алгоритм вирішення завдань, що розглядалися на практичному занятті, підготуватися до самостійного вирішення подібних завдань.
5	ТЕМА 5. Програмування арифметичних виразів. Література: [1, 6]. Завдання на СРС: Запам'ятати мету та алгоритм вирішення завдань, що розглядалися на практичному занятті, підготуватися до самостійного вирішення подібних завдань.

6	ТЕМА 6. Програмування складних арифметичних виразів. Література: [1, 6]. Завдання на СРС: Запам'ятати мету та алгоритм вирішення завдань, що розглядалися на практичному занятті, підготуватися до самостійного вирішення подібних завдань.
7	ТЕМА 7. Робота з символьним типом даних. Література: [1, 6]. Завдання на СРС: Запам'ятати мету та алгоритм вирішення завдань, що розглядалися на практичному занятті, підготуватися до самостійного вирішення подібних завдань.
8	ТЕМА 8. Робота з текстовими рядками. Частина 1. Базові операції з рядками. Література: [1, 6]. Завдання на СРС: Запам'ятати мету та алгоритм вирішення завдань, що розглядалися на практичному занятті, підготуватися до самостійного вирішення подібних завдань.
9	ТЕМА 9. Робота з текстовими рядками. Частина 2. Розширені можливості роботи з рядками. Література: [1, 6]. Завдання на СРС: Запам'ятати мету та алгоритм вирішення завдань, що розглядалися на практичному занятті, підготуватися до самостійного вирішення подібних завдань.
10	ТЕМА 10. Оператор розгалуження if/else. Оператори рівності і порівняння. Література: [1, 6]. Завдання на СРС: Запам'ятати мету та алгоритм вирішення завдань, що розглядалися на практичному занятті, підготуватися до самостійного вирішення подібних завдань.
11	ТЕМА 11. Оператор множинного вибору switch/case. Література: [1, 6]. Завдання на СРС: Запам'ятати мету та алгоритм вирішення завдань, що розглядалися на практичному занятті, підготуватися до самостійного вирішення подібних завдань.
12	ТЕМА 12. Оператори циклу. Частина 1. Цикл for. Література: [1, 6]. Завдання на СРС: Запам'ятати мету та алгоритм вирішення завдань, що розглядалися на практичному занятті, підготуватися до самостійного вирішення подібних завдань.
13	ТЕМА 13. Оператори циклу. Частина 2. Цикли while. Література: [1, 6]. Завдання на СРС: Запам'ятати мету та алгоритм вирішення завдань, що розглядалися на практичному занятті, підготуватися до самостійного вирішення подібних завдань.
14	ТЕМА 14. Оператори циклу. Частина 3. Цикл do/while. Література: [1, 6]. Завдання на СРС: Запам'ятати мету та алгоритм вирішення завдань, що розглядалися на практичному занятті, підготуватися до самостійного вирішення подібних завдань.
15	ТЕМА 15. Оператори циклу. Частина 4. Вкладені цикли та оператори розгалуження в тілі циклу. Література: [1, 6]. Завдання на СРС: Запам'ятати мету та алгоритм вирішення завдань, що розглядалися на практичному занятті, підготуватися до самостійного вирішення подібних завдань.

16	ТЕМА 16. Одномірні масиви. Частина 1. Базові операції з масивами. Література: [1, 6]. Завдання на СРС: Запам'ятати мету та алгоритм вирішення завдань, що розглядалися на практичному занятті, підготуватися до самостійного вирішення подібних завдань.
17	ТЕМА 17. Одномірні масиви. Частина 2. Принципи обробки даних в одномірних масивах. Література: [1, 6]. Завдання на СРС: Запам'ятати мету та алгоритм вирішення завдань, що розглядалися на практичному занятті, підготуватися до самостійного вирішення подібних завдань.
18	ТЕМА 18. Багатомірні масиви. Основні принципи обробки даних в багатомірних масивах. Література: [1, 6]. Завдання на СРС: Запам'ятати мету та алгоритм вирішення завдань, що розглядалися на практичному занятті, підготуватися до самостійного вирішення подібних завдань.
19	ТЕМА 19. Двомірні масиви. Основи роботи з матрицями. Література: [1, 6]. Завдання на СРС: Запам'ятати мету та алгоритм вирішення завдань, що розглядалися на практичному занятті, підготуватися до самостійного вирішення подібних завдань.
20	ТЕМА 20. Оператори циклу. Частина 4. Цикл <code>foreach</code>. Література: [1, 6]. Завдання на СРС: Запам'ятати мету та алгоритм вирішення завдань, що розглядалися на практичному занятті, підготуватися до самостійного вирішення подібних завдань.
21	ТЕМА 21. Методи. Частина 1. Розробка елементарних методів. Література: [1, 6]. Завдання на СРС: Запам'ятати мету та алгоритм вирішення завдань, що розглядалися на практичному занятті, підготуватися до самостійного вирішення подібних завдань.
22	ТЕМА 22. Методи. Частина 2. Параметри методів. Механізми передачі параметрів. Література: [1, 6]. Завдання на СРС: Запам'ятати мету та алгоритм вирішення завдань, що розглядалися на практичному занятті, підготуватися до самостійного вирішення подібних завдань.
23	ТЕМА 23. Захист практичних робіт.

Індивідуальні завдання

Варіанти практичних завдань та рекомендації щодо їх виконання зазначені у навчальному посібнику до виконання комп'ютерного практикуму.

6. Самостійна робота студента/аспіранта

№ з/п	Вид самостійної роботи	Кількість годин СРС
1	Підготовка до аудиторних занять	80
2	Підготовка до МКР	4
4	Підготовка до екзамену	30

7. Політика навчальної дисципліни (освітнього компонента)

Система вимог, які викладач ставить перед студентом:

- *правила відвідування занять: відповідно до Наказу 1-273 від 14.09.2020 р. заборонено оцінювати присутність або відсутність здобувача на аудиторному занятті, в тому числі нараховувати заохочувальні або штрафні бали. Відповідно до РСО даної дисципліни бали нараховують за відповідні види навчальної активності на лекційних та практичних заняттях. На момент проведення кожного заняття, як лекційного, так і практичного, у студента на пристрої, з якого він працює, має бути встановлено додаток Zoom (у випадку дистанційного навчання), а також відкрито курс «ОСНОВИ ПРОГРАМУВАННЯ» на платформі GoogleClassroom (код доступу до курсу надається на першому занятті згідно з розкладом). Силабус; лекційний матеріал; завдання до кожного практичного заняття; варіанти модульної контрольної роботи; екзаменаційні питання та задачі розміщено на платформі GoogleClassroom та у системі «Електронний Кампус КПІ».*
- *правила поведінки на заняттях: під час проходження курсу «ОСНОВИ ПРОГРАМУВАННЯ» студенти зобов'язані дотримуватись загальних моральних принципів та правил етичної поведінки, зазначених у Кодексі честі Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського»;*
- *правила захисту індивідуальних завдань: здобувач вищої освіти повинен своєчасно та в повному обсязі виконувати індивідуальні завдання та дати відповіді на контрольні запитання за темою свого завдання;*
- *правила призначення заохочувальних та штрафних балів: заохочувальні та штрафні бали не входять до основної шкали РСО, а їх сума не перевищує 10 % стартової шкали. Заохочувальні бали нараховують за участь у наукових конференціях, студентських конкурсах та олімпіадах. Штрафні бали не нараховуються;*
- *політика дедлайнів та перескладань: кожен здобувач вищої освіти зобов'язаний дотримуватися термінів виконання завдань у межах розкладу проведення аудиторних занять з дисципліни. Обов'язковим контрольним заходом оцінювання для допуску до екзамену є індивідуальний варіант МКР, виконання всіх індивідуальних практичних завдань та індивідуальний варіант ДКР. Здобувач вищої освіти, що з поважної причини (лікарняний, академічна мобільність тощо) не написав МКР або не виконав своєчасно індивідуальні завдання, має право зробити це під час регулярних консультацій викладача згідно розкладу. Порядок перескладання семестрового контролю визначається загальними правилами університету¹.*
- *політика щодо академічної доброчесності: Кодекс честі Національного технічного університету України «Київський політехнічний інститут» <https://kpi.ua/files/honorcode.pdf> встановлює загальні моральні принципи, правила етичної поведінки осіб та передбачає політику академічної доброчесності для осіб, що працюють і навчаються в університеті, якими вони мають керуватись у своїй діяльності, у тому числі при вивченні та складанні контрольних заходів з дисципліни «ОСНОВИ ПРОГРАМУВАННЯ». Викладачі та студенти, що вивчають дану дисципліну, зобов'язані дотримуватися положень прийнятого в університеті Кодексу честі²;*
- *при використанні цифрових засобів зв'язку з викладачем (мобільний зв'язок, електронна пошта, переписка на форумах та у соцмережах тощо) необхідно дотримуватись загальноприйнятих етичних норм, зокрема бути ввічливим та обмежувати спілкування робочим часом викладача.*

¹ Положення про поточний, календарний та семестровий контроль результатів навчання в КПІ імені Ігоря Сікорського (Додаток 1 до наказу № 7-137 від 0.08.2020 р.). URL: https://kpi.ua/document_control

² Кодекс честі Національного технічного університету України «Київський політехнічний інститут імені Ігоря Сікорського». URL: <https://kpi.ua/code>

- при виконанні практичних завдань та МКР заборонено використання штучного інтелекту.

Інклюзивне навчання. Засвоєння знань та умінь в ході вивчення дисципліни може бути доступним для більшості осіб з особливими освітніми потребами, окрім здобувачів з серйозними вадами зору, які не дозволяють виконувати завдання за допомогою персональних комп'ютерів, ноутбуків та/або інших технічних засобів.

Навчання іноземною мовою. У ході виконання завдань студентам може бути рекомендовано звернутися до англомовних джерел.

8. Види контролю та рейтингова система оцінювання результатів навчання (PCO)

Поточний контроль: захист практичної роботи (3 бали), МКР (10 балів), експрес опитування (8 балів).

Календарний контроль: провадиться двічі на семестр як моніторинг поточного стану виконання вимог силабусу. Умови проходження календарного контролю визначаються викладачем на початку семестру.

Семестровий контроль: екзамен (40 балів).

1. Рейтинг здобувача вищої освіти з кредитного модуля складається з балів, які він отримує за:

- 1) захист 14 практичних робіт;
- 2) одну МКР у вигляді тесту з 10 питань на платформі GoogleClassroom;
- 3) 8 відповідей на експрес питання на лекційних заняттях;
- 4) відповідь на екзамені, що складається з відповідей на три теоретичних питання (5 балів кожне) та вирішення однієї задачі (25 балів):

– «відмінно» – повна відповідь (не менше 90 % потрібної інформації), надані відповідні обґрунтування та особистий погляд – 35 – 40 балів;

– «добре» – достатньо повна відповідь (не менше 75 % потрібної інформації), що виконана згідно з вимогами до рівня «умінь» або містить незначні неточності – 25 – 34 балів;

– «задовільно» – неповна відповідь (не менше 60 % потрібної інформації), виконана згідно з вимогами до «стереотипного» рівня та містить деякі помилки – 15 – 24 балів;

– «незадовільно» – незадовільна відповідь – 0 – 14 балів.

Максимальна сума балів, яку здобувач вищої освіти може отримати протягом семестру $R_c = 60$ балів. Екзаменаційна складова шкали рейтингу $R_e = 40$ балів. Таким чином, рейтингова шкала з дисципліни $R = 100$ балів.

2. Критерії нарахування балів:

2.1. Захист практичної роботи: ваговий бал – 3. Максимальна кількість балів за практичну роботу – 3 бали * 14 робіт = 42 бали.

На практичних роботах студенти виконують індивідуальне завдання згідно свого варіанту.

Критерії оцінювання:

- практична робота виконана вірно та захищена протягом 2-х тижнів після виконання – 3 бали;

- практична робота виконана вірно, але захищена протягом більш ніж 2-х тижнів після виконання – 2 бали;

- практична робота виконана із незначними помилками та захищена протягом 2-х тижнів після практичного заняття – 1 бал;

- практична робота виконана із незначними помилками та захищена протягом більш ніж 2-х тижнів після практичного заняття – 0,5 бали;

- практична робота виконана із значними помилками – повертається на доопрацювання.

2.2. Написання модульної контрольної роботи: ваговий бал за одну МКР – 10. Максимальний бал за МКР складає 10 балів.

Протягом семестру проводиться одна модульна контрольна робота у вигляді тесту у середовищі Google Classroom, який складається з десяти запитань. Критерії оцінювання: 1 бал за кожну правильну відповідь.

2.3 Експрес опитування на лекційних заняттях: ваговий бал – 8. Максимальна кількість балів – 8.

Критерії оцінювання відповідей:

- повна відповідь на запитання – 1 бал;
- відповіді на запитання з деякими неточностями – 0,5 бала;
- відповіді на запитання зі значними неточностями або взагалі невірні, або студент не присутній без поважних причин на лекційному занятті – 0 балів;

3. Календарний контроль: проводиться двічі на семестр як моніторинг поточного стану виконання вимог силабусу. Умовою позитивного першого та другого календарного контролів є отримання не менше 50 % максимально можливого на момент відповідного календарного контролю рейтингу.

Умови допуску до семестрового контролю: виконання МКР, зарахування всіх практичних робіт семестровий рейтинг більше 30 балів.

Здобувачі вищої освіти, які наприкінці семестру мають рейтинг менше 30 балів, для допуску до екзамену виконують додаткову контрольну роботу.

Таблиця відповідності рейтингових балів оцінкам за університетською шкалою:

Кількість балів	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

9. Додаткова інформація з дисципліни (освітнього компонента)

Перелік тем, що виносяться на семестровий контроль:

Тема 1.1. Поняття алгоритму, його властивості, базові елементи побудови алгоритмів. Програма та предмет програмування. Мови програмування та сфери їх використання. Етапи підготовки для вирішення задачі за допомогою комп'ютера. Поняття про програмне забезпечення комп'ютера і життєвий цикл програмного забезпечення. Основні етапи реалізації програм: створення вихідного коду, компіляція та інтерпретація, виконання.

Тема 1.2. Ключові слова, ідентифікатори, оператори. Змінні. Визначення змінних. Вирази, операції і оператори. Інтегроване середовище розробки Visual Studio. Структура простої програми C#. Коментарі та їх використання.

Тема 2.1. Базові операції консольного введення і виведення. Форматування виведення. Форматування числових даних.

Тема 2.2. Системні типи даних. Оголошення і ініціалізація змінних. Внутрішні типи даних і операція new. Вирази та арифметичні оператори.

Тема 2.3. Числові типи даних. Логічний тип даних. Символьний тип даних. Типи даних для роботи з датами та часом. Поняття неявно типізованих змінних.

Тема 2.4. Рядки. Робота із рядковими даними. Базові маніпуляції рядками. Конкатенація рядків. Порівняння рядків.

Тема 3.1. Базові алгоритмічні структури: слідування, розгалуження, повторення. Оператор розгалуження `if / else`. Оператори рівності і порівняння. Оператор множинного вибору `switch`.

Тема 3.2. Ітераційні конструкції. Цикл `for`. Цикл `foreach`. Цикли `while` і `do / while`. Алгоритми з використанням вкладених циклів з складеними операторами.

Тема 3.3. Методи (підпрограми). Методи як засіб структурної декомпозиції програм. Параметри методів. Механізми передачі параметрів. Передача параметрів за значенням і за посиланням. Типи і аргументи методів. Модифікатори `ref` і `out`.

Тема 3.4. Масиви. Ініціалізація масивів. Організація доступу до елементів масиву. Заповнення масивів за допомогою генератора випадкових чисел. Принципи обробки даних в одномірних масивах. Методи сортування та пошуку даних.

Тема 3.5. Робота з багатомірними масивами. Використання масивів у якості аргументів і значень. Основні принципи використання класу `System.Array`.

Перелік питань для екзамену з навчальної дисципліни «ОСНОВИ ПРОГРАМУВАННЯ»:

1. Електронно-обчислювальна машина: визначення, архітектура.
2. Процесор: принцип роботи та основні характеристики.
3. Архітектура процесорів з програмної точки зору.
4. Операційна система: визначення, призначення та основні функції.
5. Арифметичні основи ЕОМ. Цифра. Число. Система числення: визначення. Найбільш використовувані в обчислювальній техніці системи числення.
6. Подання інформації в ЕОМ. Алгоритм: визначення, основні вимоги та форми запису.
7. Алгоритмічні мови: призначення, основні види. Компілятор. Транслятор.
8. Програма: визначення. Базові складові мови програмування C#: алфавіт, визначення та призначення зарезервованих слів.
9. Ідентифікатор: визначення та правила синтаксису. Визначення літерала.
10. Коментарі: призначення, види та синтаксис.
11. Основне завдання програми. Змінні та константи: визначення, область дії.
12. Основні типи даних в мові C#.
13. Типи даних за замовчуванням. Призначення суфіксів при присвоєнні значень змінним.
14. Операції консольного введення/виведення. Використання плейсхолдерів. Навести приклад.
15. Перетворення типів: призначення, приклади.
16. Вирази в мові C#. Операції та операнди.
17. Арифметичні операції в мові C#. Унарні та бінарні операції. Асоціативність операторів.
18. Порозрядні логічні операції. Операції зсуву.
19. Операції присвоювання. Складені оператори присвоювання.
20. Операції порівняння. Логічні операції.
21. Математичні функції класу `Math`.
22. Символьний тип даних в мові C#. Опис та основні можливості.
23. Рядковий тип даних в мові C#. Опис та основні можливості.
24. Структурне програмування: сутність, причини виникнення та принципи.
25. Оператори розгалуження `if`, `if...else`.
26. Оператор множинного вибору `switch`.
27. Тернарний оператор.
28. Цикли: основні визначення, класифікація та правила вибору.
29. Цикл по лічильнику в мові C#.
30. Цикл з передумовою в мові C#.
31. Цикл з післяумовою в мові C#.
32. Масиви: основні визначення, поняття та види масивів.
33. Одновимірні масиви в мові C#. Процедура створення одновимірних масивів.
34. Ініціалізація одновимірних масивів. Звернення до елементів простого масиву.
35. Цикл `foreach`.
36. Багатовимірні масиви в мові C#: створення та основні характеристики.

37. Ініціалізація багатовимірних масивів. Звернення до елементів багатовимірного масиву.
38. Зубчасті масиви в мові C#: створення та основні характеристики.
39. Ініціалізація зубчастих масивів. Звернення до елементів зубчастого масиву.
40. Властивості класу `System.Array`.
41. Сортуння та пошук значень в одновимірному масиві.
42. Зміна розміру одновимірного масиву. Перевернення одновимірного масиву.
43. Списки: призначення та створення.
44. Ініціалізація списку.
45. Основні властивості класу `List<T>`. Пошук елемента в списку.
46. Переліки: основні визначення та створення.
47. Звернення до елементів переліку.
48. Особливості використання перетворення типів для елементів переліку.
49. Методи: визначення, синтаксис та основні правила створення.
50. Виклик методу. Повернення значення та вихід з методу.
51. Передача параметрів методу: параметри-значення, параметри-посилання.
52. Іменовані аргументи та опціональні параметри методу.
53. Вихідні параметри методу.
54. Локальні функції.

Перелік задач для екзамену з навчальної дисципліни «ОСНОВИ ПРОГРАМУВАННЯ»:

1. Написати програму визначення, якій чверті координатної площини належить точка з координатами x , y . Вважати, що точка не може лежати на осях ($x \neq 0$, $y \neq 0$). Пояснити, як вона працює.
2. Написати програму визначення, яка конвертує оцінку за п'ятибальною шкалою з числа в текстове представлення з використанням оператора `switch`. Пояснити, як вона працює.
3. Написати програму підсумовування парних цілих чисел від 0 до 20 з використанням оператора `for`. Пояснити, як вона працює.
4. Написати програму, яка перевіряє, чи є в масиві {4, 7, 13, 20, 33, 23, 54} число кратне 13-ти з використанням операторів `for` та `break`. При знаходженні такого числа, перервати виконання циклу. Пояснити, як вона працює.
5. Написати програму, яка знаходить суму непарних цілих чисел від 0 до 20 з використанням операторів `for`, `if` та `continue`. Пояснити, як вона працює.
6. Написати програму піднесення числа 2 до низку ступенів від 1 до 10 з використанням оператора `while`. Результати вивести на екран консолі. Пояснити, як вона працює.
7. Написати програму, що розраховує факторіал числа, що не дорівнює 0, з використанням оператора `do...while`. Пояснити, як вона працює.
8. Написати програму, що створює одновимірний масив з п'яти елементів типу `int`, проініціалізувати кожен елемент вручну за допомогою індексатору. Вивести значення на екран за допомогою будь-якого циклу. Пояснити, як вона працює.
9. Написати програму, що створює два одновимірних масиви з трьома елементами типу `int` в одному та `string` в другому за допомогою неявної типізації. Вивести на екран консолі тип кожного масиву за допомогою методу `GetType()`. Пояснити, як вона працює.
10. Написати програму, що створює одновимірний масив з чотирьох об'єктів типу `bool`, `int`, `string`, `float`. За допомогою циклу `foreach` та методу `GetType()` вивести на екран консолі тип кожного елементу створеного масиву об'єктів. Пояснити, як вона працює.
11. Написати програму, що створює одновимірний масив з п'яти елементів типу `int`, проініціалізувати кожен елемент вручну за допомогою індексатору. Вивести значення на екран за допомогою циклу `foreach`. Пояснити, як вона працює.
12. Написати програму, що створює двовимірний масив 4×5 типу `int`, проініціалізувати кожен елемент випадковими цілими числами від 1 до 15 за допомогою класу `Random()`. Вивести значення на екран за допомогою вкладених циклів `for`. Пояснити, як вона працює.

13. Написати програму, що створює зубчастий масив з трьох елементів типу `int`: перший – одномірний масив з двох елементів типу `int`; другий – одномірний масив з трьох елементів типу `int`; третій – одномірний масив п'яти елементів типу `int`. Вручну проініціалізувати кожен елемент цілими числами за допомогою індексатору. Вивести значення на екран за допомогою будь-якого циклу. Пояснити, як вона працює.
14. Написати програму, що створює двомірний масив `10x10` типу `int`, проініціалізувати кожен елемент випадковими цілими числами від 1 до 15 за допомогою класу `Random()`. Вивести значення на екран за допомогою вкладених циклів `for`. Вивести на екран властивості масиву `Length` та `Rank`. Пояснити, як вона працює.
15. Написати програму, що створює одномірний масив з двох елементів типу `string` за допомогою методу `CreateInstance()`, проініціалізувати кожен елемент за допомогою методу `SetValue()` та вивести значення одного з елементів масиву на екран, використовуючи метод `GetValue()`. Пояснити, як вона працює.
16. Написати програму, що створює багатовимірний масив за допомогою методу `CreateInstance()` та одновимірного масиву розмірів `{2, 3, 4, 5}`. Вивести на екран властивості масиву `Length` та `Rank`. Пояснити, як вона працює.
17. Написати програму, що копіює 5 елементів масиву `{-4, -3, -2, -1, 0, 1, 2, 3, 4}`, починаючи з 2-го індексу, в інший масив за допомогою методу `Array.Copy()`. Вивести на екран елементи вихідного та скопійованого масиву за допомогою будь-якого циклу. Пояснити, як вона працює.
18. Написати програму, що здійснює сортування масиву `{4, 5, -183, 12, 34, 0, 2, -13}` та бінарний пошук у відсортованому масиві числа 12 за допомогою методів `Array.Sort()` та `Array.BinarySearch()`. Вивести на екран елементи вихідного та відсортованого масиву за допомогою будь-якого циклу та результати пошуку. Пояснити, як вона працює.
19. Написати програму, що здійснює сортування 5-ти елементів масиву `{4, 5, -183, 12, 34, 0, 2, -13}`, починаючи з 2-го індексу та бінарний пошук у відсортованому масиві числа 12 за допомогою методів `Array.Sort()` та `Array.BinarySearch()`. Вивести на екран елементи вихідного та відсортованого масиву за допомогою будь-якого циклу та результати пошуку. Пояснити, як вона працює.
20. Написати програму, що здійснює пошук у масиві `{4, 5, -183, 12, 34, 12, 2, -13}` індексу першого та останнього входження числа 12 за допомогою методів `Array.IndexOf()` та `Array.LastIndexOf()`. Вивести на екран елементи вихідного масиву за допомогою будь-якого циклу та результати пошуку. Пояснити, як вона працює.
21. Написати програму, що здійснює зміну розміру одновимірного масиву з семи елементів типу `string`: збільшення на 5 елементів та зменшення до 4-х елементів за допомогою методу `Array.Resize()`. Вивести на екран елементи вихідного масиву та отриманих масивів за допомогою будь-якого циклу. Пояснити, як вона працює.
22. Написати програму, що здійснює перевернення одновимірного масиву з семи елементів типу `string` за допомогою методу `Array.Reverse()`. Вивести на екран елементи вихідного масиву та перевернутого масиву за допомогою будь-якого циклу. Пояснити, як вона працює.
23. Написати програму, що створює список типу `string` початковою ємністю 100 з початковою ініціалізацією 5-ти елементів. Вивести на екран загальну кількість елементів списку, фактичну кількість елементів списку та елемент за індексом 3, за допомогою властивостей `Capacity`, `Count`, `Item[Int32]`. Пояснити, як вона працює.
24. Написати програму, що створює шестивимірний масив типу `int` зі значеннями вимірів `[6,5,4,3,2,1]` (ініціалізувати масив не потрібно). Вивести на екран індекси першого, останнього елементу кожного виміру та довжину кожного виміру масиву за допомогою методів `GetLowerBound()`, `GetUpperBound()`, `GetLength()`. Пояснити, як вона працює.
25. Написати програму, що створює список типу `string` з початковою ініціалізацією 5-ти елементів. Додати елемент в список, вставити елемент в список на будь-яку позицію та провести пошук будь-якого елементу, що входить до списку, за допомогою методів `Add()`, `Insert()` та `Contains()`. Вивести на екран початковий список та отриманий список за допомогою циклу `foreach`. Пояснити, як вона працює.

26. Написати програму, що створює перелік TrafficLight з трьох констант, що визначають три стану світлофору (Red, Yellow, Green). Вивести на екран ім'я константи Green, використовуючи синтаксис звернення до елементів переліку та значення самої константи. Пояснити, як вона працює.

27. Написати програму, де визначено метод SayHello(), який виводить на екран рядок «Hello, World». Вивести на екран цей рядок за допомогою виклику цього методу. Пояснити, як вона працює.

28. Написати програму, що створює перелік TrafficLight з трьох констант, що визначають три стану світлофору (Red, Yellow, Green). Вивести на екран ім'я всіх трьох констант по їх значення, використовуючи перетворення типів для елементів переліку. Пояснити, як вона працює.

29. Написати програму, де визначається метод Add(), що повертає суму двох чисел, які передаються йому, як параметри за значенням. Вивести на екран доданки та результат. Пояснити, як вона працює.

30. Написати програму, де визначено два локальних метода, які повертають максимальне та мінімальне значення двох чисел. Вивести на екран результати роботи програми. Пояснити, як вона працює.

Для самостійного вивчення пропонуються відносно прості запитання, які носять у більшості випадків описовий характер, призначені для розширення кругозору студентів та повторення матеріалів, які вивчались у інших дисциплінах, та мають безпосереднє відношення до дисципліни.

Методи та форми навчання включають не лише традиційні університетські лекції та семінарські зайняття, а також елементи роботи в командах та групових дискусій. Застосовуються стратегії активного навчання, які визначаються такими методами та технологіями: методи проблемного навчання (дослідницький метод); особистісно-орієнтовані технології, засновані на таких формах і методах навчання як кейс-технологія і проектна технологія; візуалізація та інформаційно-комунікаційні технології, зокрема електронні презентації для лекційних занять. Комунікація з викладачем будується за допомогою використання інформаційної системи «Електронний кампус», платформи дистанційного навчання «Сікорський» на базі G Suite for Education, а також такими інструментами комунікації, як електронна пошта і WhatsApp. Під час навчання та для взаємодії зі студентами використовуються сучасні інформаційно-комунікаційні та мережеві технології для вирішення навчальних завдань.

Робочу програму навчальної дисципліни (силабус):

Складено доцент кафедри електропостачання НН ІЕЕ, к.т.н., Філянін Данило Володимирович

Ухвалено кафедрою електропостачання НН ІЕЕ (протокол № 22 від 24 червня 2026 р.).

Погоджено Навчально-методичною комісією НН ІЕЕ (протокол № 43 від 29 червня 2026 р.).