

СИЛАБУС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Назва дисципліни	ОСНОВИ ПРОГРАМУВАННЯ
Рівень вищої освіти	Перший (бакалаврський)
Форма здобуття освіти	Денна
Семестр	I
Обсяг дисципліни	кредити – 3,0 загальний обсяг годин – 90 лекцій – 10
Форма звітності	Залік
Викладачі	Командирчик Андрій Васильович, викладач
Контактна інформація	Телефон: +38 (096) 885-29-51 E-mail: andreikomandirchuck@gmail.com

ЗАГАЛЬНИЙ ОПИС ДИСЦИПЛІНИ

Навчальна дисципліна «Основи програмування» спрямована на формування у студентів базових знань і навичок у галузі програмування. Курс охоплює основні концепції, принципи та методи програмування, які дозволять студентам створювати прості програми на популярних мовах програмування, а також розвивати логічне мислення та алгоритмічні навички.

МЕТА НАВЧАННЯ

Основними цілями курсу є: – формування фундаментальних знань про програмування як галузь; – розуміння основних принципів та конструкцій програмування; – набуття практичних навичок написання кодів і розробки програм; – розвиток умінь у створенні алгоритмів та вирішенні практичних завдань.

Очікувані результати навчання: знати: – основні поняття програмування та його етапи; – базові конструкції мов програмування (умовні оператори, цикли, функції); – поняття змінних, типів даних та структур даних; – принципи написання простих програм і їх тестування.

вміти: – розробляти алгоритми для розв'язання простих задач; – писати код на вибраній мові програмування (наприклад, Python, Java, C++); –

використовувати базові бібліотеки та інструменти для розробки програмного забезпечення; – проводити налагодження та тестування програмного коду..

ТЕМАТИЧНИЙ ПЛАН

1. Вступ до програмування
 - 1.1. Що таке програмування?
 - 1.2. Історія розвитку мов програмування.
 - 1.3. Основні поняття: програма, алгоритм, мова програмування.
2. Алгоритми та структури даних
 - 2.1. Визначення та типи алгоритмів.
 - 2.2. Основні структури даних: масиви, списки, словники.
 - 2.3. Алгоритми сортування та пошуку.
3. Основи синтаксису мов програмування
 - 3.1. Змінні та типи даних.
 - 3.2. Умовні оператори (if, switch).
 - 3.3. Цикли (for, while).
4. Функції та модульність
 - 4.1. Що таке функція?
 - 4.2. Параметри та повернення значень.
 - 4.3. Модульність програм та використання бібліотек.
5. Обробка помилок та налагодження
 - 5.1. Основи налагодження програмного коду.
 - 5.2. Обробка виключень та помилок.
 - 5.3. Тестування програм.
6. Сучасні інструменти програмування
 - 6.1. Системи контролю версій (Git).
 - 6.2. IDE та їх функціональні можливості.
 - 6.3. Основи роботи з документацією.

ВИДИ РОБІТ І ЗАВДАНЬ. ІНСТРУМЕНТАРІЙ НАВЧАННЯ

Курс включає лекційні та лабораторні заняття, а також самостійну роботу. Більшу частину балів студенти отримують за засвоєння теоретичного матеріалу

на лекціях, виконання лабораторних завдань, підготовку звітів з виконаних робіт, захист проектів та індивідуальні завдання.

Для виконання лабораторних робіт знадобиться комп'ютер з встановленою операційною системою (Windows, macOS або Linux), а також:

– Інстальоване програмне забезпечення для розробки (IDE, наприклад, PyCharm, Visual Studio Code).

– Мова програмування (Python, Java, C++), залежно від вибраної програми курсу.

Інструменти для налагодження та тестування програмного забезпечення.

ПОЛІТИКА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Відвідування занять є одним із способів накопичення балів. Проте, значна увага приділятиметься насамперед самостійній роботі, під час якої здобувачі освіти матимуть більше можливостей розширити, поглибити та удосконалити знання й уміння, набуті під час аудиторних занять та консультацій чи самостійного опанування адаптованих викладачем навчальних матеріалів – конспектів лекцій, методичних роз'яснень щодо ходу виконання практичних завдань. Формами звітності за самостійну роботу виступатимуть звіти з результатами виконаних завдань. Вимоги до самостійної роботи студента включають систематичність, вчасність та якість виконання завдань. Здобувачі освіти повинні дотримуватися встановлених термінів здачі робіт та виконувати настанови викладача щодо їх структури та формату.

Для забезпечення політики академічної доброчесності, студентам необхідно самостійно виконувати завдання, уникаючи плагіату, фальсифікації даних та списування. Використання інформаційних джерел повинно бути належним чином оформлене згідно з вимогами наукового стилю та стандартами цитування. В разі виявлення порушень академічної доброчесності можуть застосовуватися відповідні санкції, включаючи зниження оцінки або навіть скасування результатів роботи.

СИСТЕМА ОЦІНЮВАННЯ

Студент може отримати до 45 балів за виконання й захист завдань лабораторного практикуму, до 25 балів за виконання й захист завдань самостійної роботи і до 30 балів під час модульного контролю.

Оцінка за кожну лабораторну та самостійну роботу включає такі складові:

- виконання (наскільки повно й правильно виконані завдання роботи);
- захист (наскільки вільно студент володіє матеріалом і здатен представити результати виконаної роботи);
- звіт (наскільки грамотно, якісно й повно задокументовано у звіті результати виконаної роботи);
- дотримання графіка виконання (студент одержить максимальні бали за цією складовою за умови вчасного виконання роботи; у випадку відхилень від графіка кількість балів може знижуватися).