

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ВОДНОГО ГОСПОДАРСТВА ТА
ПРИРОДОКОРИСТУВАННЯ**

Навчально-науковий інститут кібернетики, інформаційних технологій та інженерії

04-04-87S

СИЛАБУС		SYLLABUS	
Програмування		Programming	
Шифр за ОП	ОК 13	Code in Degree Programme	
Освітній рівень: Бакалаврський (перший)		Level of Education: Bachelor's (first)	
Галузь знань Інформаційні технології	12	Field of Knowledge Information Technology	
Спеціальність Комп'ютерна інженерія	123	Field of Study Computer Engineering	
Освітня програма: Комп'ютерна інженерія		Degree Programme: Computer Engineering	

Силабус навчальної дисципліни «Програмування» для здобувачів вищої освіти ступеня «бакалавр», які навчаються за освітньо-професійною програмою «Комп'ютерна інженерія», спеціальності 123 «Комп'ютерна інженерія». Рівне. НУВГП. 2025. 17 стор.

ОП на сайті університету:
<https://ep3.nuwm.edu.ua/30336/>

Розробники силабусу: Сидор Андрій Іванович,
к.т.н., в.о. завідувача кафедри
обчислювальної техніки,
Бойчура Михайло
Володимирович, к.т.н., доцент
кафедри обчислювальної техніки

Силабус схвалений на засіданні кафедри
Протокол № 9 від "21" лютого 2025 року

В.о. завідувача кафедри: Сидор А.І., к.т.н.

Керівник (гарант) ОП: Сидор А.І., к.т.н.

Схвалено науково-методичною радою з якості ННІ КІТІ
Протокол № 4 від "24" лютого 2025 року

Голова науково-методичної ради з якості
ННІ: Мартинюк П. М., д.т.н., професор.

Попередня версія силабусу: –

© Сидор А.І., 2025

© Бойчура М.В., 2025

© НУВГП, 2025

ПРОГРАМА ОСВІТНЬОГО КОМПОНЕНТА	
Програмування	
ЗАГАЛЬНА ІНФОРМАЦІЯ	
Ступінь вищої освіти	бакалавр
Освітня програма	Комп'ютерна інженерія
Спеціальність	123 Комп'ютерна інженерія
Рік навчання, семестр	1-й рік, 1-й семестр 1-й рік, 2-й семестр
Групи	KI-11 KI-21фб KI3-21фб

Кількість кредитів	1 семестр 2 семестр	4,5 5,5	– 6,5	– 6,5
Лекції:	1 семестр 2 семестр	28 22	– 28	– 4
Лабораторні заняття:	1 семестр 2 семестр	24 28	– 38	– 16
Самостійна робота:	1 семестр 2 семестр	83 115	– 129	– 175
Курсова робота:	так			
Форма навчання	денна/заочна			
Форма підсумкового контролю	екзамен			
Мова викладання	державна			

ІНФОРМАЦІЯ ПРО РОЗРОБНИКА (ІВ)

Лектор		Сидор Андрій Іванович к.т.н., в.о. завідувача кафедри обчислювальної техніки.
Вікіситет	https://wiki.nuwm.edu.ua/index.php/Сидор Андрій Іванович	
ORCID	https://orcid.org/0000-0003-4911-7034	
Як комунікувати	a.i.sydor@nuwm.edu.ua	
Лектор		Бойчуря Михайло Володимирович к.т.н., доцент кафедри обчислювальної техніки.
Вікіситет	https://wiki.nuwm.edu.ua/index.php/Бойчуря Михайло Володимирович	
ORCID	https://orcid.org/0000-0002-9073-4037	
Як комунікувати	m.v.boichura@nuwm.edu.ua , групи у месенджерах	

ІНФОРМАЦІЯ ПРО ОСВІТНІЙ КОМПОНЕНТ

Мета та завдання

Програмування є ключовою складовою сучасних інформаційних технологій. У цьому контексті мова С++ займає особливе місце завдяки своїй продуктивності, широким можливостям та гнучкості. Вона використовується у розробці операційних систем, ігрових рушіїв, високопродуктивного програмного забезпечення та багатьох інших сфер. Опанування С++ є базовим кроком, який дозволить у перспективі створювати ефективні та масштабовані програми, що відповідають сучасним вимогам індустрії.

Дисципліна «Програмування» покликана сформувати у студентів міцний фундамент знань та навичок, необхідних для написання якісного та ефективного коду. Протягом курсу студенти ознайомляться з базовими концепціями програмування, структурою програм, принципами об'єктно-орієнтованого програмування (ООП) та інструментами для командної розробки. Крім того, особлива увага приділяється розробці практичних навичок, необхідних для професійної діяльності в галузі програмування.

Метою дисципліни є формування у студентів базових знань та практичних навичок з програмування мовою С++, розробки графічних інтерфейсів, реалізації мережевих додатків та командної взаємодії.

Завдання дисципліни:

1. Ознайомити студентів з основними концепціями та поняттями у програмуванні на прикладі мови С++.
2. Засвоїти методи організації коду і структурування програм, а також принципи написання ефективного і підтримуваного коду.
3. Надати знання про основні конструкції мови С++.
4. Ознайомити студентів з основами об'єктно-орієнтованого програмування.
5. Дослідити питання використання шаблонів та перевантаження операторів у С++.
6. Вивчити принципи та підходи до написання якісного програмного забезпечення: патерни проектування, UML-діаграми, SOLID, DRY, KISS, YAGNI.
7. Надати навички роботи з інструментами контролю версій та основами командної розробки програмного забезпечення.
8. Ознайомити студентів з методами роботи з файлами, обробкою помилок та основами сокетного програмування.
9. Вивчити основи розробки графічного інтерфейсу за допомогою Windows Forms у С++.
10. Сприяти закріпленню знань через виконання командного довгострокового проєкту, що моделює завдання програмної інженерії.

Посилання на розміщення освітнього компонента на навчальній платформі Moodle, на платформі освітніх програм та їхніх освітніх компонентів

<https://exam.nuwm.edu.ua/course/view.php?id=5667>

Передумови вивчення*

(місце освітнього компоненту в структурно-логічній схемі)

Дисципліни, що передують вивченню: дисципліна вивчається одночасно з ОК 8 «Вища математика».

Отримані навички можуть використовуватись при подальшому вивченні дисциплін: ОК 14 «Практична підготовка з програмування», ОК 25 «Системне програмування», ОК 26 «Об'єктно-орієнтоване програмування», ВБ 1.1 «Web-програмування», ВБ 1.2 «Основи Web-безпеки», ВБ 2.2 «Кібербезпека апаратно-програмних засобів».

Компетентності

Z2. Здатність вчитися і оволодівати сучасними знаннями.

Z3. Здатність застосовувати знання у практичних ситуаціях.

Z5. Здатність спілкуватися іноземною мовою.

Z8. Здатність працювати в команді.

P2. Здатність використовувати сучасні методи і мови програмування для розроблення алгоритмічного та програмного забезпечення.

P3. Здатність створювати системне та прикладне програмне забезпечення комп'ютерних систем та мереж.

P7. Здатність використовувати та впроваджувати нові технології, включаючи технології розумних, мобільних, зелених і безпечних обчислень, брати участь в модернізації та реконструкції комп'ютерних систем та мереж, різноманітних вбудованих і розподілених додатків, зокрема з метою підвищення їх ефективності.

Програмні результати навчання (ПРН). Результати навчання (РН)*

N5. Мати знання основ економіки та управління проектами.

N6. Вміти застосовувати знання для ідентифікації, формулювання і розв'язування технічних задач спеціальності, використовуючи методи, що є найбільш придатними для досягнення поставлених цілей.

N7. Вміти розв'язувати задачі аналізу та синтезу засобів, характерних для спеціальності.

N8. Вміти системно мислити та застосовувати творчі здібності до формування нових ідей.

N10. Вміти розробляти програмне забезпечення для вбудованих і розподілених застосувань, мобільних і гібридних систем, розраховувати, експлуатувати, типове для спеціальності обладнання.

N12. Вміти ефективно працювати як індивідуально, так і у складі команди.

N16. Вміти оцінювати отримані результати та аргументовано захищати прийняті рішення.

N21. Спілкуватись усно та письмово з професійних питань українською мовою та однією з іноземних мов (англійською, німецькою, італійською, французькою, іспанською).

N24. Усвідомлювати необхідність навчання впродовж усього життя з метою поглиблення набутих та здобуття нових фахових знань, удосконалення креативного мислення.

Структура та зміст навчальної дисципліни

№	Теми (лекції)	Опис лекції	№	Теми лабораторних занять
МОДУЛЬ 1. Основи програмування				

1	Вступ до програмування (2 год.) N5, N10, N24	Введення в мови програмування C і C++. Введення в розробку програмного забезпечення. Встановлення IDE. Вирішення найбільш поширених проблем початківців в C++.	1	Ввід та вивід інформації (2 год.)
1	Загальна структура програми (2 год.) N5, N6, N8, N16, N24	Налаштування компілятора: попередження та помилки; вибір стандарту мови C++. Структура програми. Коментарі. Змінні, ініціалізація і присвоювання. Об'єкти cout, cin та endl. Функції. Параметри та аргументи функцій. Локальна область видимості. Ключові слова та ідентифікатори.	2	Використання математичних операцій (2 год.)
1	Основи C++ (2 год.) N5, N8	Оператори. Базове форматування коду. Прототип функції і попереднє оголошення. Багатофайлові програми. Заголовкові файли. Директиви препроцесора. Header guards і директива #pragma once. Конфлікт імен і простір імен std.	3	Використання функцій (2 год.)
1	Змінні та основні типи даних в C++ (2 год.) N6, N10, N16	Відлагодження програм: степпінг і точки зупинки. Ініціалізація, присвоювання та оголошення змінних. Тип даних void. Розміри типів даних. Цілочисельні типи даних: short, int та long.	4	Базове форматування коду (2 год.)
1	Типи даних у C++ (2 год.) N6, N12, N16, N24	Фіксований розмір цілочисельних типів даних. Типи даних з плаваючою комою: float, double і long double. Логічний тип даних bool. Символьний тип даних char. Літерали і магичні числа. const, constexpr і символічні константи.	5	Робота з декількома функціями (2 год.)
1	Оператори в C++ (2 год.) N5, N24	Пріоритет операцій і правила асоціативності. Арифметичні оператори. Інкремент, декремент і побічні ефекти. Умовний тернарний оператор, оператор sizeof і кома. Оператори порівняння. Логічні оператори: I, АБО, НЕ.	6	Використання операторів та різних типів даних (2 год.)

1	Область видимості в C++ (2 год.) N8, N10, N16	Побітові оператори. Бітові флаги та бітові маски. Блоки стейтментів. Глобальні змінні. Статичні змінні. Зв'язки, область видимості та тривалість життя. Простір імен. using-стейтменти. Неявна конвертація типів даних.		
МОДУЛЬ 2. Основні конструкції в програмуванні				
1	Інші типи змінних в C++ (2 год.) N8, N10, N16, N24	Явна конвертація типів даних. Введення в std::string. Перелічуваний тип даних. Класи enum. Псевдоніми типів: typedef і type alias. Структури. Вивід типів: ключове слово auto. Оператори управління потоком виконання програм.	7	Робота з рядками (2 год.)
1	Цикли і розгалуження в C++ (2 год.) N6, N8, N10, N16, N24	Оператори умовного розгалуження if/else. Оператор switch. Оператор goto. Цикл while. Цикл do while. Цикл for. Оператори break і continue. Генерація рандомних чисел. Обробка некоректного користувацького вводу.	8	Використання циклів та розгалужень (2 год.)
1	Масиви та Рядки в C++ (2 год.) N6, N10, N16, N24	Масиви. Фіксовані масиви. Масиви і цикли. Двовимірні масиви. Рядки C-style. Клас std::string_view.	9	Робота з масивами (2 год.)
1	Вказівники в C++ (2 год.) N6, N7, N8, N10, N16, N24	Вказівники. Нульові вказівники. Вказівники і масиви. Адресна арифметика та індексація масивів. Символьні константи рядків C-style. Динамічне виділення пам'яті. Динамічні масиви.	10	Робота з динамічними масивами (2 год.)
1	Посилання в C++ (2 год.) N6, N7, N8, N10, N16, N24	Посилання. Посилання і const. Оператор доступу до членів через вказівник. Цикл foreach. Вказівники на вказівники. Введення в std::array та std::vector (вектори). Введення в літератори. Алгоритми в стандартній бібліотеці C++.	11	Робота з масивами та функціями (2 год.)

1	Функції в C++ (2 год.) N6, N7, N8, N10, N16, N24	Параметри та аргументи функцій. Передача по значенню. Передача по посиланню. Вбудовані функції.	12	Командна розробка програм з використанням функцій (2 год.)
1	Робота з функціями в C++ (2 год.) N6, N7, N8, N10, N12, N16, N24	Перевантаження функцій. Параметри за замовчуванням. Вказівники на функції. Стек і Купа. Ємність вектора. Рекурсія і Числа Фібоначчі.		
МОДУЛЬ 3. Вступ до об'єктно-орієнтованого програмування				
1	Структури, класи та об'єкти. Відлагодження програм (2 год.) N6, N7	Структури. Класи. Об'єкти. Конструктори та деструктори. Інструменти відлагодження.	13	Робота з декількома користувацькими класами (2 год.)
1	Шаблони. Перевантаження операторів (2 год.) N6, N24	Основи шаблонів функцій та класів. Перевантаження операторів.	14	Користувацькі класи з використанням шаблонів та перевантаження операторів (2 год.)
1	Принципи об'єктно-орієнтованого програмування. Патерни проектування (4 год.) N6, N7, N8, N16, N21, N24	Інкапсуляція, наслідування, поліморфізм, абстракція. Абстрактні класи та їх використання. Віртуальні функції. Динамічний поліморфізм. Віртуальні деструктори. Модифікатори доступу. Патерни Singleton та Observer.	15	Розробка додатку згідно принципів об'єктно-орієнтованого програмування (4 год.)
1	UML-діаграми. Принципи SOLID, DRY, KISS, YAGNI (2 год.) N5, N6, N7, N8, N12, N16, N21, N24	UML-діаграми класів. UML-діаграми класів у контексті SOLID. Детальний розбір принципів SOLID, DRY, KISS та YAGNI. Рефакторингу коду. Користувацькі динамічні бібліотеки.	16	Проектування архітектури додатку (2 год.)
МОДУЛЬ 4. Командна розробка мережевих Desktop-додатків				
1	Контроль версій та командна робота (2 год.) N5, N6, N12, N21, N24	Основи Git та GitHub. Робота з гілками, Pull Request. Використання Jira для керування проектами.	17	Системи контролю версій та проектів (2 год.)
1	Робота з файлами у C++. Обробка помилок (2 год.) N16	Потоки введення/виведення. Читання та запис у файли. Бінарні файли. Види помилок у C++. Використання try-catch.	18	Запис та читання текстових і бінарних файлів (2 год.)

1	Windows Forms у C++ (4 год.) N10, N24	Розробка складних графічних інтерфейсів. Події та обробники подій. Інтерактивні елементи керування. Взаємодія з файлами та мережею у Windows Forms.	19	Побудова складних графічних інтерфейсів (4 год.)
1	Основи сокетного програмування (4 год.) N6, N7, N8, N16, N10, N12, N21, N24	Основи клієнт-серверної архітектури. Використання TCP та UDP у C++. Сокетного програмування (Boost.Asio).	20	Розробка клієнт-серверного додатку на основі сокетного з'єднання (4 год.)
1	Командне довгострокове завдання (6 год.) N5, N6, N7, N8, N10, N12, N16, N21, N24	GitHub. Jira. Командна робота. Мережеве програмування. Розробка Desktop-додатків. Презентація проєктів. Написання резюме.	21	Командне довгострокове завдання з розробки мережевого застосунку з графічним інтерфейсом (14 год.)
Форми, методи та технології навчання				

Форми навчання	<ul style="list-style-type: none"> • очна (денна); • заочна.
Форми навчального процесу	<ul style="list-style-type: none"> • навчальні заняття (лекції, лабораторні заняття, консультації); • самостійна робота здобувачів; • робота в наукових бібліотеках та мережі Інтернет; • контрольні заходи (поточна складова оцінювання, модульні контролі, підсумковий контроль).
Методи технології та навчання	<ul style="list-style-type: none"> • демонстрація; • навчальна дискусія; • дебати; • робота в малих групах (команді) та індивідуальна робота; • проєктна технологія; • аналіз конкретних ситуацій (case study): ситуація-оцінка; • контекстне навчання; • проблемне навчання.
Процес навчання включає, зокрема, наступне	<ul style="list-style-type: none"> • написання комп'ютерних програм; • відлагодження програм; • Code Review; • слідування рекомендаціям Coding Conventions та принципу Big Design Up Front.
Засоби навчання	<ul style="list-style-type: none"> • відеозапис лекції; • презентація; • конспект лекцій; • різні туторіали.

Інструменти, обладнання, програмне забезпечення

- середовище розробки: Microsoft Visual Studio 2022 (з компонентами для розробки консольних і десктопних додатків на мові C++);
- Git термінал: GitHub Desktop;
- система керування проєктами Jira.

Порядок оцінювання програмних результатів навчання/ результатів навчання

Студент може отримати 100 балів, враховуючи наступну розбаловку:

- модульні контрольні роботи (40 балів):
 - 1-й модульний контроль 20 балів (рівень 1 – 18 запитань по 0.5 балів за правильну відповідь, рівень 2 – 10 питань по 0.6 балів за правильну відповідь, рівень 3 – 5 запитань по 1 балу за правильну відповідь);
 - 2-й модульний контроль 20 балів (рівень 1 – 18 запитань по 0.5 балів за правильну відповідь, рівень 2 – 10 питань по

0.6 балів за правильну відповідь, рівень 3 – 5 запитань по 1 балу за правильну відповідь);

- лабораторні роботи (60 балів):

- перший семестр: 14 балів за відвідування лекційних занять, 3 бали за правильне виконання лабораторної роботи (1 бал – виконання лабораторної роботи; 1 бал – усний захист лабораторної роботи та 1 бал – виконання індивідуального завдання по темі лабораторної роботи) та 10 балів за вчасний захист лабораторних робіт;

- другий семестр: за теми лабораторних робіт 13-20 передбачено по 5 балів, тоді як за тему №21 – 20 балів; у випадку правильного виконання лабораторної роботи оцінка лінійно залежить від відсотка розуміння коду; за певних умов можливе повне перенесення Story Points, вказаних у Jira, в електронний журнал; як альтернатива, студенти можуть виконувати завдання з використанням інших мов чи технологій програмування, але за умови попереднього узгодження деталей з викладачем;

- додаткові бали (до 10 балів) за волонтерство, олімпіаду, спартакіади, конкурси, конференції, написання статей, активна студентська діяльність, конкретні пропозиції з удосконалення змісту навчальної дисципліни тощо.

Основні критерії, що характеризують рівень компетентності здобувача вищої освіти при оцінюванні результатів поточного та підсумкового контролю з навчальної дисципліни:

- виконання всіх видів навчальної роботи, що передбачені силабусом навчальної дисципліни;

- глибина і характер знань навчального матеріалу за змістом навчальної дисципліни, що міститься в основних та додаткових рекомендованих літературних джерелах;

- вміння аналізувати явища, що вивчаються, у їх взаємозв'язку і розвитку;

- характер відповідей на поставлені запитання (чіткість, лаконічність, логічність, послідовність тощо);

- вміння застосовувати теоретичні положення під час розв'язання практичних задач;

- вміння аналізувати достовірність одержаних результатів;

- своєчасність виконання;

- дотримання вимог до оформлення (технологічної документації, ДСТУ тощо).

Критерії оцінювання лабораторних робіт (у % від кількості балів, виділених на завдання із заокругленням до цілого числа):

- 0% – завдання не виконано або студент не може пояснити запрограмований код;

- 40% – завдання виконано частково та містить суттєві помилки методичного або розрахункового характеру, порушені терміни виконання та вимоги до оформлення;

- 60% – завдання виконано повністю, але містить суттєві помилки у розрахунках або в методиці, порушені терміни виконання та вимоги до оформлення;

- 80% – завдання виконано повністю і вчасно, проте містить окремі несуттєві недоліки (висновки, оформлення тощо);
- 100% – завдання виконано правильно, вчасно і без зауважень.

Рекомендована література (основна, допоміжна)

Основна література

1. Рудий Т.В., Паранчук Я.С., Сенік В.В. Алгоритмізація та програмування. Частина 1. Структурне програмування: навчальний посібник. Львів : Львівський державний університет внутрішніх справ, 2023. 240 с.
2. Stroustrup B. *Programming: Principles and Practice Using C++ (C++ In-depth)*. 3rd ed. Hoboken : Addison-Wesley Professional, 2024. 656 p.
3. Зеленський О.С., Лисенко В.С. Об'єктно-орієнтоване програмування на C++: навч. посіб. Кривий Ріг : Державний університет економіки і технологій, 2023. 215 с.
4. Фратавчан В.Г., Фратавчан Т.М., Лазорик В.В. Алгоритмізація та програмування: навч. посіб. Чернівці : ЧНУ, 2022. 286 с.
5. Глинчук Л.Я., Гришанович Т.О. Програмування: підручник. Луцьк : ВНУ ім. Лесі Українки, 2022. 160 с.

Допоміжна література

1. Гришанович Т.О., Глинчук Л.Я. Основи об'єктно-орієнтованого програмування: навч. посіб. Луцьк : ВНУ імені Лесі Українки, 2022. 120 с.
2. Водка О.О., Дашкевич А.О., Іванченко К.В. та ін. Основи програмування на C++: навч. посіб. Харків : НТУ «ХПІ», 2021. 112 с.
3. Ментинський С.М., Пелех Я.М. Основи програмування на C++: навч. посіб. Львів : Галицька Видавнича Спілка, 2021. 256 с.
4. Stroustrup B. *Tour of C++, A (C++ In-Depth Series)*. 3rd ed. Hoboken : Addison-Wesley Professional, 2022. 320 p.
5. Bancila M. *Modern C++ Programming Cookbook: Master Modern C++ with comprehensive solutions for C++23 and all previous standards* 3rd ed. Birmingham : Packt Publishing, 2024. 816 p.

Інформаційні ресурси в Інтернеті

1. C++ Core Guidelines. URL: <https://isocpp.github.io/CppCoreGuidelines/CppCoreGuidelines> (Last accessed: 12.11.2024).
2. C++ Теорія – YouTube. URL: <https://www.youtube.com/playlist?list=PLiPRE8VmJzOpn6PzYf0higmCEyGzo2A5g> (Last accessed: 12.11.2024).
3. Уроки C++ для початківців / Програмування на C++ з нуля – YouTube. URL: <https://www.youtube.com/playlist?list=PL7vq4D0vOpQa9WaLe7btV01eixBUZ6-Ve> (Last accessed: 12.11.2024).

ПОЛІТИКИ ВИКЛАДАННЯ ТА НАВЧАННЯ

Перелік соціальних, «м'яких» навичок (soft skills)

<p>Вміння комунікувати.</p>	<ul style="list-style-type: none"> • здатність спілкуватися державною мовою як усно, так і письмово; • вміння спілкуватись та писати із використанням англomовної професійної термінології; • навички усного спілкування; • навички письмового спілкування; • вміння писати зрозумілий код.
<p>Вміння сумісно працювати.</p>	<ul style="list-style-type: none"> • навички управління проєктами; • здатність планувати свій час у сенсі співставлення вимог, власних знань, здібностей і дедлайнів; • здатність працювати в команді; • навички міжособистісних відносин; • вміння надавати рекомендації іншим у коректній формі.
<p>Здатність до аналізу та синтезу.</p>	<ul style="list-style-type: none"> • вміння критично мислити; • здатність знаходити вихід із складних ситуацій; • здатність до навчання; • вміння комплексного вирішення проблем.
<p>Здатність застосовувати знання у практичних ситуаціях.</p>	

Дедлайни та перескладання

Захист лабораторної роботи здійснюється на поточному занятті. У випадку пропуску заняття можна відпрацювати лабораторну роботу, оформивши відповідний звіт. Прийом звітів та їх поточне оцінювання завершується на останньому підсумковому занятті.

На здачу кожного з модульних контролів студенту надається одна спроба. Перший модульний контроль здається на будь-якій лекції у жовтні (квітні), а другий – на передостанній чи останній парі згідно розкладу. Перездача окремого модульного контролю передбачена лише за виключних обставин.

У випадку, якщо студент здав обидва модулі та набрав сумарно хоча б 60 балів, то він має право отримати дану підсумкову оцінку «автоматом»; в усіх інших випадках студент повинен йти на іспит.

У разі, якщо по закінченню сесії здобувач не набрав 60 балів, його відправляють на комісію для розгляду результатів підсумкового контролю, яка рекомендує здобувачу із академічною заборгованістю покращити результати підсумкового контролю одним із наступних шляхів: повторного складання підсумкового контролю, повторного вивчення освітньої компоненти чи повторного курсу навчання.

Неформальна та інформальна освіта

Здобувачі мають право на часткове або повне перезарахування дисципліни за умови написання ними відповідної заяви, заповнення декларації та надання документів, які підтверджують ті результати навчання, які здобувач отримав (див. положення <https://ep3.nuwm.edu.ua/28363/>). Зокрема студенти можуть самостійно проходити онлайн-курси на таких навчальних платформах, як Prometheus, Coursera, edX, edEra, FutureLearn та інших, для наступного перезарахування результатів навчання. Проте доцільно попередньо узгодити з викладачем відповідність обраного онлайн-курсу суті навчальної дисципліни. Деякий перелік підходящих курсів наведено нижче:

- Coursera – Programming in C++: A Hands-on Introduction Specialization (Програмування на C++: практичний вступ до спеціалізації);

- Coursera – Object Oriented Programming Specialization (Спеціалізація об'єктно-орієнтованого програмування);

- Coursera – Introductory C Programming Specialization (Вступний курс з програмування мовою C).

Зручний пошук курсів доступний тут: <https://www.classcentral.com/>.

Окрім того, якщо з'являються обставини для здобуття неформальної чи інформальної освіти від викладачів-практиків, то пропонуються ці можливості для студентів; рекомендуються відео-уроки практикуючих програмістів з YouTube тощо.

Правила академічної доброчесності

Задля запобігання академічної недоброчесності вимагається наступне:

- кожен студент повинен виконувати власне індивідуальне завдання;
- кожна студентська команда повинна виконувати власний проєкт;
- кожне завдання у проєкті студент виконує самостійно, відмічаючи це у Jira та GitHub;
- студенти повинні розуміти програмний код виконаного завдання;
- студентам забороняється: плагіятити, самоплагіятити, фабрикувати, фальсифікувати, списувати, обманювати тощо.

Залежно від виду та ступеня порушення викладач може накладати наступні санкції:

- усне або письмове зауваження від викладача;
- попередження про можливість притягнення до академічної відповідальності;
- зниження чи анулювання результатів оцінювання навчального завдання здобувача вищої освіти;
- повторне виконання навчального завдання;
- виконання іншого навчального завдання;
- призначення додаткового навчання з питань академічної доброчесності;
- призначення додаткових контрольних заходів (додаткові індивідуальні навчальні завдання, тести тощо);
- подання клопотання на ім'я ректора з метою порушення формальної процедури розгляду питання про притягнення студента до відповідальності.

За списування під час проведення модульного контролю чи підсумкового контролю студент позбавляється подальшого права здавати матеріал і у нього виникає академічна заборгованість.

Документи стосовно академічної доброчесності (про плагіат, порядок здачі курсових робіт, кодекс честі студентів, документи Національного агентства стосовно доброчесності) наведені на сторінці «Якість освіти» сайту НУВГП – <http://nuwm.edu.ua/sp/akademichna-dobrochesnistj>.

Вимоги до відвідування

Пропущені заняття відпрацьовуються здобувачами самостійно. Електронні варіанти лекцій доступні на платформі Moodle та/або в групах в месенджерах; лабораторні роботи виконуються у вигляді звіту та захищаються на наступному занятті чи на консультації.

Відвідування пари допускається із використанням власного ноутбука. Студенти не повинні порушувати дисципліну на парі.

Для студентів, які знаходяться на індивідуальному плані навчання, надаються індивідуальні завдання.

Автор
Доцент ОТ

Михайло БОЙЧУРА

Затверджено

Проректор з науково-педагогічної та
навчальної роботи

Валерій СОРОКА



документ підписаний КЕП
Номер документа СИЛ №770
Підписувач Сорока Валерій Степанович
Підписувач (дані КЕП):
Сертифікат 3FAA9288358EC003040000009B6C3700C8C2C100