Міністерство освіти і науки України
Національний аерокосмічний університет
«Харківський авіаційний інститут»

Кафедра математичного моделювання та штучного інтелекту (№ 304)

ЗАТВЕРДЖУЮ

Гарант освітньої програми

_____ Дмитро ЧУМАЧЕНКО
(підпис) (ім'я та ПРІЗВИЩЕ)

«_21_» _____08_____ 2025 р.

СИЛАБУС ОБОВ'ЯЗКОВОЇ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Основи програмування (мова С++)
(назва навчальної дисципліни)

Галузь знань: F «Інформаційні технології»
(шифр і найменування галузі знань)

Спеціальність: F3 «Комп'ютерні науки»
(код і найменування спеціальності)

Освітня програма: «Інтелектуальні системи та технології»
(найменування освітньої програми)

Рівень вищої освіти: ***перший (бакалаврський)***

Силабус уведено в дію з 01.09.2025

Харків – 2025 р.

Розробник: Коробчинський К.П., доц. каф. 304, к. т. н., доц.
(прізвище та ініціали, посада, науковий ступінь і вчене звання) (підпис)

Силабус навчальної дисципліни розглянуто на засіданні кафедри № 304
математичного моделювання та штучного інтелекту
(назва кафедри)

Протокол № 1 від « 21 » серпня 2025 р.

В. о. завідувача кафедри к.ф.-м.н., доцент Олексій КАРТАШОВ
(науковий ступінь і вчене звання) (підпис) (ім'я та ПРІЗВИЩЕ)

Погоджено з представником здобувачів освіти:
Здобувач першого (бакалаврського)
рівня вищої освіти за ОП
«Інтелектуальні системи та технології» Катерина ПИЛИПЕНКО
(підпис) (ім'я та ПРІЗВИЩЕ)

1. Загальна інформація про викладача

ПІБ: *Коробчинський Кирил Петрович*

Посада: *доцент каф. 304*

Науковий ступінь: *канд. техн. наук*

Вчене звання: *доцент*

Перелік дисциплін, які викладає:
Програмування та алгоритмічні мови; Системне програмування; Бази даних та інформаційні системи; Організація баз даних; Нереляційні бази даних; Git для розподіленої розробки ПЗ; Використання Git для розробки програмного забезпечення; Web-програмування; Розподілені інформаційно-аналітичні системи

Напрями наукових досліджень:
Програмування. Бази даних. ШІ

Контактна інформація:
k.korobchinskiy@khai.edu

2. Опис навчальної дисципліни

Форма здобуття освіти	Денна
Семестр	1-2
Мова викладання	Українська
Тип дисципліни	Обов’язкова
Обсяг дисципліни: кредити ЄКТС/ кількість годин	7+5,5 кредитів ЄКТС / 375 годин (176 аудиторних, з яких: лекції – 80, лабораторні – 96; СРЗ – 199);
Види навчальної діяльності	Лекції та лабораторні заняття, самостійна робота
Види контролю	Поточний контроль, модульний контроль, семестровий контроль – іспит
Пререквізити	“Англійська мова”, “Інформатика”, ”Математична логіка”

3. Мета та завдання навчальної дисципліни, переліки компетентностей та очікуваних результатів навчання

Мета вивчення: викладення основних понять алгоритмізації і техніки застосування у програмуванні базових алгоритмічних структур (організація програм) і базових структур даних (організація даних).

Для досягнення мети поставлені такі **основні завдання:**

- вивчення основних етапів процесу проектування програмного забезпечення і визначення принципів процедурного програмування щодо розробки програм мовою С++;
- вивчення типових підходів до розробки і аналізу найбільш розповсюджених алгоритмів рішення економіко-математичних задач;
- здійснення аналізу можливостей сучасних інструментальних середовищ розробки програм (на прикладі середовища Visual Studio.NET);
- визначення концепцій і вивчення основних принципів організації програм у ОС Windows.

Предметом навчальної дисципліни є теорія і практика застосування у програмуванні базових алгоритмічних структур і базових структур даних на базі сучасних технологій розробки програмного забезпечення.

Дисципліна "Програмування та алгоритмічні мови" є базовою для бакалаврів галузі знань "Інформаційні технології".

Згідно з вимогами освітньо-професійної програми студенти повинні досягти таких **компетентностей:**

ЗК-1 Здатність до абстрактного мислення, аналізу та синтезу

ЗК-2 Здатність застосовувати знання у практичних ситуаціях

ЗК-3 Здатність планувати і управляти часом

ЗК-4 Знання та розуміння предметної області та розуміння професійної діяльності

ЗК-7 Здатність до пошуку, оброблення та аналізу інформації з різних джерел

ФК-1 Здатність використовувати системний аналіз як сучасну міждисциплінарну методологію, що базується на прикладних математичних методах та сучасних інформаційних технологіях і орієнтована на вирішення задач аналізу і синтезу технічних, економічних, соціальних, екологічних та інших складних систем.

ФК-2 Здатність формалізувати проблеми, описані природною мовою, у тому числі за допомогою математичних методів, застосовувати загальні підходи до математичного моделювання конкретних процесів.

ФК-8 Здатність організовувати роботу з аналізу та проектуванню складних систем, створення відповідних інформаційних технологій та програмного забезпечення.

Програмні результати навчання:

ПРН2 Вміти використовувати стандартні схеми та методи для розв’язання обчислювальних, комбінаторних та логічних задач, що сформульовані природною мовою, застосовувати класичні алгоритми для перевірки властивостей та класифікації об’єктів, множин, відношень, графів, груп, кілець, решіток, булевих функцій, тощо.

ПРН8 Володіти сучасними методами розробки програм і програмних комплексів, та прийняття оптимальних рішень щодо складу програмного забезпечення, алгоритмів, процедур і операцій.

ПРН9 Вміти створювати ефективні алгоритми для обчислювальних задач системного аналізу та систем підтримки прийняття рішень.

ПРН16 Знати та вміти застосовувати методи управління складними процесами й системами на основі нечіткої логіки, а також визначати межі можливих застосувань математичних моделей і методів.

Міждисциплінарні зв’язки:

Необхідна навчальна база перед початком вивчення дисципліни: для успішного вивчення дисципліни необхідні базові знання, отримані студентами в об’ємі шкільної програми, а також поточні знання при паралельному освоєнні ними дисциплін "Іноземна мова", "Алгебра та геометрія", "Математичний аналіз", "Теорія алгоритмів і математична логіка", "Архітектура обчислювальних систем".

У результаті вивчення навчальної дисципліни студенти повинні:

знати:

- основні етапи процесу проектування програмного забезпечення;
- типові алгоритмічні конструкції;
- принципи процедурного і структурованого програмування;
- особливості застосування сучасних базових інструментальних програмних засобів, призначених для вирішення економічних задач;
- базові типи даних;
- похідні типи даних: переліки, покажчики, посилання, масиви, структури, об’єднання;
- оператори управління програмою;
- команди перед процесорної обробки;
- правила роботи з функціями;
- систему уведення-виведення C++;
- основні принципи роботи з файлами;
- правила роботи із шаблонами;
- принципи розробки Windows-додатків;
- основу побудови програм на керованому C++;

вміти: складати програми мовою C++, забезпечуючи:

- рішення задач з курсу вищої математики;
- створення і обробку структур, масивів структур; найпростішу обробку файлів;
- використовування функцій;

- використовування основних елементів призначеного для користувача інтерфейсу;
- використовування сучасного інструментального програмного забезпечення;
- користуватися раніше складеними програмами і здійснювати супровід програм, вносити зміни в програму, виконувати відладку програм за допомогою вбудованих інструментальних засобів.

мати уявлення:

- про основні етапи створення об'єктно-орієнтованого програмного додатку;
- про фундаментальні принципи класичного об'єктно-орієнтованого підходу до програмування комп'ютерної системи;
- про сучасні способи використання мови програмування С++.

4. Зміст навчальної дисципліни

Модуль 1. Введення в розробку і кодування алгоритмів.

Змістовий модуль 1. Комп'ютер з точки зору програміста

Тема 1. Організація пам'яті та організація даних.

Обчислювальна машина: процесор, пам'ять, периферійні пристрої; принцип дії. Організація пам'яті, особливості її використання. Комірка, адреса, ємність, розподіл пам'яті. Мова ЕОМ, команда, операція, операнди. Організація даних: біти, байти, слова, розташування у пам'яті. Програмне забезпечення ЕОМ, склад і призначення. Операційні системи(ОС).

Тема 2. Файли та типи виконання програм.

Файли. Етапи виконання програм: трансляція, редагування, виконання. Типи модулів.

Змістовий модуль 2. Основи алгоритмізації.

Тема 3. Основи алгоритмізації.

Поняття алгоритму. Властивості алгоритму. Типові алгоритмічні конструкції. Розробка алгоритму методом покрокового уточнення. Алгоритмічні конструкції: послідовність, вибір, повторення. Способи завдання алгоритму. Критерії оцінки алгоритмів.

Початкові відомості про технологію програмування: процедурне, структуроване і об'єктно-орієнтоване програмування. Мови програмування: процедурні, аплікативні, системи правил, об'єктно-орієнтовні. Стандартизація мов та середовища проектування. Транслятори. Редактори. Компонувальники. Відладчики. Керуючі структури: оператори, вирази та підпрограми. Огляд сучасних інтегрованих систем програмування. Інтегроване середовище системи програмування Visual Studio.NET. Платформа .NET. Етапи розробки та впровадження програм. Вимоги до програмного коду.

Алгоритми, властивості алгоритмів. Виконавці алгоритмів. Загальні правила алгоритмічної мови. Величини. Вказівка присвоювання. Алгоритми з розгалуженнями. Етапи розв'язування задач.

Тема 4. Алгоритмічна декомпозиція.

Алгоритмізація типових обчислювальних задач.Поняття алгоритму та типові алгоритмічні структури програмування. Базові структури алгоритмів. Побудова алгоритмів. Основні риси та характеристики алгоритмів.

Модуль 2. Розробка лінійних алгоритмів.

Змістовий модуль 3. Основи програмування алгоритмічною мовою.

Тема 5. Інструменти і базові засоби програмування.

Шаблон програми. Оператори та вирази. Типи даних. Поняття змінної. Створення лінійних програм. Розв'язування задач на створення лінійних програм. Структура С++ програми.

Лексичні елементи мови С++: алфавіт, коментарі, ідентифікатори, службові слова, дані, вираз, операнд, змінна, операція. Домовленості про імена.

Поняття типу даних. Класифікація і представлення даних. Базові типи даних: логічний, символьний, цілий, речовинний. Перетворення типів: неявні перетворення, явні перетворення.

Пріоритети операції. Зведена таблиця пріоритетності і асоціативності операцій.

Тема 6. Мова С/С++ та її елементи.

Мова С/С++. Елементи мови: множина символів, дані та їх характеристики. Константи . Змінні, їх атрибути. Проста змінна, її типи та опис. Вказівники. Ініціалізація даних. Структура простої програми .

Операції. Унарні операції: унарний мінус, унарний плюс, порозрядне інвертування, логічне заперечення, інкремент, декремент, операція обчислення розміру (sizeof). Бінарні операції: адитивні, мультиплікативні, зсувів, порозрядні, операції відносин, логічні, привласнення.

Пріоритети операції. Зведена таблиця пріоритетності і асоціативності операцій.

Стандартні математичні функції.

Константні величини: цілі, речовинні, перечислювальні, символьні (літерні), рядкові (рядки або літерні рядки). Правила визначення компілятором констант. Визначення констант за допомогою ключового слова const. Константи переліків.

Тема 7. Операції та вирази.

Первинні вирази. Унарні та бінарні вирази. Вирази присвоювання. Операції: послідовного присвоювання, умовна та явного перетворення типів. Пріоритет та асоціативність операцій.

Змістовий модуль 4. Розробка алгоритмів з розгалуженнями.

Тема 8. Оператори управління програмою. Команди та дані.

Лінійний операторний процес та операторний процес з розгалуженням. Умовні оператори та оператор вибору. Специфіка роботи з масивами. Основні типи задач на масиви даних: вибірка, сортування, фільтрування, обхід. Вирази,

символи пропусків, блоки і комплексні вирази. Операнд, змінна. Оператор привласнення. Оголошення та ініціалізація змінних.

Загальні положення. Оператор: позначений, порожній, складений, умовний, вкладений умовний. Оператор - вираз. Оператор перемикач. Оператор завершення. Оператор безумовного переходу.

Тема 9. Оператори циклу.

Типи операторів. Найпростіший оператор, оператор-оголошення, оператор-визначення, оператор-вираз. Управляючі оператори: оператори проходження; оператори вибору (єдиний вибір – if, подвійний вибір – if / else, множинний вибір – switch, умовна операція); оператори повторення (оператор while, оператор do-while, оператор for). Вкладені цикли. Управляючі оператори в циклах: оператор break, оператор continue, оператор goto. Рекомендації по вибору циклів.

Приклади використання операторів. Передача керування у циклах. Оператор continue.

Тема 10. Програмна реалізація.

Програмна реалізація алгоритмів різної структури. Приклади створення програм.

Тема 11. Налагоджування програм.

Помилки. Етапи налагоджування: контроль правильності програми, локалізація помилок, їх виправлення. Крековий режим налагодження на ПК. Приклад.

Тема 12. Комплексний тип даних.

Програмна реалізація комплексного типу даних. Програмування операцій . Стандартна бібліотека COMPLEX.H та її основні функції. Приклади програм обробки інформації із комплексними даними. Поняття покажчика, посилання. Покажчики і масиви. Адресна арифметика. Посилання. Приклади використовування покажчиків і посилань.

Покажчики на функції. Посилання. Параметри функцій як посилання. Організація пам’яті в сучасних процесорах і покажчики мови С++. Моделі пам’яті. Статичні і динамічні змінні. Оператори new і delete. Дина- мічні масиви. Динамічні масиви як параметри функцій. Зв’язні списки.

Модуль 3. Процедурне програмування. Рядки.

Змістовий модуль 5. Організація похідних типів даних та алгоритми їх оброблення.

Тема 13. Абстракція даних. Складені структури даних.

Поняття основних структур даних. Основні характеристики базових структур даних. Алгоритми для обробки базових структур даних.

Тема 14. Масиви. Масиви, їх опис, ініціалізація.

Масиви. Оголошення масивів та розміщення у пам’яті. Ініціалізація масивів. Обробка одновимірних масивів даних економічного характеру. Алгоритми сортування масивів. Багатовимірні масиви. Ініціалізація багатовимірного масиву. Типові приклади обробки матриць.

Тема 15. Обробка символьної інформації.

Символьний тип даних, символьна константа. Рядки, константа-рядок. Бібліотека string.h функцій обробки рядків. Введення-виведення рядків та масивів символів. Розташування у пам'яті. Обробка рядків: різноманітний пошук у рядку, вставлення та видалення символів, аналіз слів, копіювання та поєднання рядків. Алгоритми та демонстраційні програми роботи з текстами. Рядки як масиви символів. Операції з рядками. Тип даних string.

Тема 16. Файли.

Загальні відомості про систему вводу-виводу даних. Текстовий файл. Типи файлів і оголошення їх змінних. Відкриття файлу для запису /читання . Вказівник файлу Переміщення вказівника. Розміщення даних у файлі . Запис у файл. Читання із файлу. Закриття файлу.

Тема 17. Моделі та класи пам'яті.

Моделі пам'яті. Класи пам'яті. Змінні глобальні ,локальні, автоматичні, розміщення у пам'яті. Особливості використання пам'яті. Вказівники на різні об'єкти. Робота з вказівниками. Приклади типових використань.

Тема 18. Змінні різних типів пам'яті.

Змінні різних типів пам'яті у багатофайлових програмах. Приклади.

Змістовий модуль 6. Розробка функцій та їх типи.

Тема 19. Функції.

Загальні відомості про функції. Структура функції. Значення, параметри і аргументи, що повертаються. Оголошення функції. Прототипи функцій. Визначення функції. Виконання функції. Локальні і глобальні змінні. Правило видимості змінних. Приведення типів аргументів функцій. Правила автоматичного (неявного) приведення типів. Явні перетворення типів. Правила роботи з функціями. Класи пам'яті. Список параметрів функції. Параметри за умовчанням. Способи передачі параметрів. Способи повернення значення. Функції, що підставляються.

Створення власних заголовних файлів. Перевантаження функцій. Рекурсія. Робота функцій. Розбиття пам'яті. Стек і функції. Модифікатори функцій.

Тема 20. Робота з функціями.

Масиви як параметри функцій. Робота з функціями.

Типові алгоритми і їх реалізація. Приклади .

Тема 21. Базові засоби процедурного програмування.

Процедурно-орієнтоване програмування. Рекурсія.

Процедури, методи, підпрограми, функції. Поняття процедурно-орієнтованого програмування. Рекурсія в програмуванні. Приклади рекурсії.

Тема 22. Особливості процедурного програмування на базі С++.

Методології розробки програм. Використання процедур. Процедурний підхід.

Процедурна декомпозиція. Особливості процедурного програмування. Використання умовних операторів і оператора безумовного переходу.

Основні методології розробки програм. Низхідне та висхідне проектування, модульне програмування.

Тема 23. Шаблони

Основи апарату шаблонів. Шаблони функцій. Перевантаження шаблонів функцій. Шаблони функцій сортування.

Стандартна бібліотека шаблонів (STL). Призначення та склад STL. Контейнери. Робота з векторами.

Модуль 4. Введення в ООП.

Змістовий модуль 7. Структурні типи даних та робота з ними.

Тема 24. Структурні типи даних.

Структури, об'єднання, перелічення: їх опис, розміщення у пам'яті, ініціалізація, використання у програмах.

Структури. Структури з бітовими полями. Вкладені структури. Доступ до елементів структур. Операції з структурами. Структури як параметри функцій.

Масиви структур. Покажчики на структури. Передача з посилання масивів структур. Об'єднання. Операції з об'єднаннями. Переліки. Функції роботи з датою та часом.

Тема 25. Основи об'єктно-орієнтованого підходу до створення програм.

Парадигми програмування: об'єктно-орієнтована парадигма (інкапсуляція, поліморфізм, наслідування).

Тема 26. Основні властивості ООП.

Реалізація об'єктно-орієнтованого підходу до програмування засобами мови програмування С++.

Класи і їх опис в С++. Екземпляри класів або об'єкти. Поля (атрибути) та методи. Специфікатори доступу (private, protected, public). Вказівник this. Перевантаження методів. Конструктори і деструктори класів. Перевантаження операцій. Наслідування в С++

Тема 27. Відносини та зв'язки між класами

Основи UML. Сфера застосування. Основні поняття та принципи. Зв'язки між класами (асоціація, агрегація, композиція, узагальнення).

UML: діаграми класів. UML: діаграми взаємодії. UML: діаграми послідовності. Шаблони проектування.

5. Методи навчання

1. За джерелами придбання знань – словесні: лекція (вступна, традиційна, проблемна, з помилками), бесіда (евристична), диспут, дискусія, робота з друкованими та інтернет-джерелами; наочні: ілюстрація, спостереження; практичні: вправа, лабораторна робота.

2. За характером пізнавальної діяльності тих, хто навчається – інформаційно-репродуктивний, репродуктивний, проблемне викладання, частково-пошуковий.
3. За логікою пізнання – індуктивний, дедуктивний, аналогій, вивідних знань.
4. Методи перевірки й оцінки знань, умінь, навичок: спостереження, усне опитування, контрольні роботи, програмований контроль, тестування (традиційне та машинне).

6. Методи контролю

Визначення рівня засвоєння студентом навчального матеріалу дисципліни здійснюється шляхом проведення поточних (захист лабораторних робіт) і підсумкових контролів (захист змістовного модуля, екзамену).

Поточний контроль має на меті систематично відстежувати, наскільки студент розуміє та засвоює матеріал, вміє виконувати лабораторні роботи, самостійно вирішувати завдання та писати відповідний код, готувати звіти та представляти результати дослідження в усній або письмовій формі.

Підсумковий контроль покликаний оцінити глибину знань студента з усього курсу, його розуміння зв'язків між різними темами, здатність застосовувати отримані знання на практиці та самостійно вирішувати задачі з аналізу даних, які виникають в рамках дисципліни.

Таким чином методами контролю є:

1. Опитування.
2. Лабораторні роботи.
3. Модульні контрольні роботи.

Форма підсумкового контролю успішності навчання: іспит.

7. Критерії оцінювання та розподіл балів, які отримують здобувачі освіти

Таблиця 7.1 – Розподіл балів, які отримують здобувачі освіти

Складові навчальної роботи	Бали за одне заняття (завдання)	Кількість занять (завдань)	Сумарна кількість балів
Модуль 1			
Виконання і захист лабораторних (практичних) робіт	0...5	5	0...25
Модульний контроль	0…15	1	0…15
Модуль 2			
Робота на лекціях	0…1	5	0…5
Виконання і захист лабораторних (практичних) робіт	0...5	4	0...20
Модульний контроль	0…15	1	0…15
Виконання і захист РГР (РР, РК)	0…20	1	0…20
Усього за семестр1			**0…100**
Модуль 3			
Робота на лекціях	0…1	5	0…5
Виконання і захист лабораторних (практичних) робіт	0...5	4	0...20
Модульний контроль	0…15	1	0…15
Модуль 4			
Робота на лекціях	0…1	5	0…5
Виконання і захист лабораторних (практичних) робіт	0...5	4	0...20
Модульний контроль	0…15	1	0…15
Виконання і захист РГР (РР, РК)	0…20	1	0…20
Усього за семестр2			**0…100**

Семестровий контроль (іспит/залік) проводиться у разі відмови студента від балів поточного тестування й за наявності допуску до іспиту/заліку. Під час складання семестрового іспиту/заліку студент має можливість отримати максимум 100 балів.

Білет для іспиту/заліку складається з 4 *теоретичних та практичних питання та максимальна кількість балів за кожне питання 25 балів (сума – 100 балів).*

7.2. Якісні критерії оцінювання

Необхідний обсяг **знань** для одержання позитивної оцінки:

• основні етапи процесу проектування програмного забезпечення;
• типові алгоритмічні конструкції;
• принципи процедурного і структурованого програмування;
• особливості застосування сучасних базових інструментальних програмних засобів, призначених для вирішення економічних задач;
• базові типи даних;
• похідні типи даних: переліки, покажчики, посилання, масиви, структури, об'єднання;
• оператори управління програмою;
• правила роботи з функціями;
• систему уведення-виведення С++;
• основні принципи роботи з файлами;

Необхідний обсяг **вмінь** для одержання позитивної оцінки:

- складати програми мовою С++, забезпечуючи:
- рішення задач з курсу вищої математики;
- створення і обробку структур, масивів структур; найпростішу обробку файлів;
- використовування функцій;
- використовування сучасного інструментального програмного забезпечення.

7.3 Критерії оцінювання роботи студента протягом семестру

Задовільно (60-74). Показати мінімум знань та умінь. Захистити всі індивідуальні завдання та здати тестування. Знати основні етапи процесу проектування програмного забезпечення, типові алгоритмічні конструкції, принципи процедурного і структурованого програмування; особливості застосування сучасних базових інструментальних програмних засобів, призначених для вирішення економічних задач. Уміти складати програми мовою С++, забезпечуючи: рішення задач з курсу вищої математики, створення і обробку структур, масивів структур; найпростішу обробку файлів.

Добре (75-89). Твердо знати мінімум, захистити всі індивідуальні завдання, виконати всі КР , здати тестування та поза аудиторну самостійну роботу. Уміти використовувати похідні типи даних: переліки, покажчики, посилання, масиви, структури, об'єднання. Володіти операторами управління програмою та використовувати як базові типи даних, так і розроблені користувачем. Знати правила роботи з функціями та основні принципи роботи з файлами. Вміти використовувати функції, основні елементи призначеного для користувача інтерфейсу.

Відмінно (90-100). Здати всі контрольні точки з оцінкою «відмінно». Досконально знати всі теми та уміти застосовувати їх.

Шкала оцінювання: бальна і традиційна

Сума балів	Оцінка за традиційною шкалою	
	Іспит, диференційований залік	Залік
90 – 100	Відмінно	Зараховано
75 – 89	Добре	
60 – 74	Задовільно	
0 – 59	Незадовільно	Не зараховано

8. Політика навчального курсу

Відвідування занять. З огляду на інтерактивний формат курсу, відвідування практичних занять є обов'язковим. Студенти, які не можуть регулярно відвідувати заняття, повинні протягом тижня узгодити з викладачем індивідуальний графік відпрацювання пропущених занять. Пропущені заняття відпрацьовуються на найближчій консультації протягом тижня після пропуску у формі співбесіди за питаннями з плану заняття. В окремих випадках дозволяється відпрацювання у письмовій формі шляхом виконання індивідуального письмового завдання.

Дотримання вимог академічної доброчесності здобувачами освіти під час вивчення навчальної дисципліни. Під час вивчення навчальної дисципліни здобувачі освіти мають дотримуватися загальноприйнятих морально-етичних норм і правил поведінки, вимог академічної доброчесності, передбачених Положенням про академічну доброчесність Національного аерокосмічного університету «Харківський авіаційний інститут» (*https://khai.edu/assets/files/polozhennya/polozhennya-pro-akademichnu-dobrochesnist.pdf*). Очікується, що роботи здобувачів освіти будуть їх оригінальними дослідженнями або міркуваннями. Відсутність посилань на використані джерела, фабрикування джерел, списування, втручання в роботу інших здобувачів освіти становлять, але не обмежують, приклади можливої академічної недоброчесності. Виявлення ознак академічної недоброчесності в письмовій роботі здобувача освіти є підставою для її незарахування викладачем незалежно від масштабів плагіату чи обману.

Вирішення конфліктів. Порядок і процедури врегулювання конфліктів, пов’язаних із корупційними діями, зіткненням інтересів, різними формами дискримінації, сексуальними домаганнями, міжособистісними стосунками та іншими ситуаціями, що можуть виникнути під час навчання, а також правила етичної поведінки регламентуються Кодексом етичної поведінки в Національному аерокосмічному університеті «Харківський авіаційний інститут» (*https://khai.edu/ua/university/normativna-baza/ustanovchi-dokumenti/kodeks-etichnoi-povedinki/*).

9. Методичне забезпечення

Електронний ресурс, на якому розміщено навчально-методичний комплекс дисципліни – http://library.khai.edu/library/fulltexts/doc/_002__Programuvannya.pdf, який включає в себе:

- конспект лекцій, підручники (навчальні посібники), в тому числі в електронному вигляді, які за змістом повністю відповідають робочій програмі дисципліни;
- методичні вказівки та рекомендації для виконання курсових робіт та проектів, розрахункових та розрахунково-графічних робіт, лабораторних та практичних робіт, а також рекомендації для самостійної підготовки;

- тематики індивідуальних завдань;
- приклади розв'язування типових задач чи виконання типових завдань;
- питання, тести для контрольних заходів;
- каталоги інформаційних ресурсів.

Увесь науково методичний комплект з дисципліни розміщено на освітньому порталі: *https://study.korobchinskiy.com/course/view.php?id=8*

10. Рекомендована література

Базова

1. Коробчинський К.П., Москович І.В., Пічугіна О.С. Програмування та алгоритмічні мови : методичний посібник – Харків : Нац. аерокосм. ун-т ім. М. Є. Жуковського "Харків. авіац. ін-т", 2021. – 160с.

2. Коробчинський К.П., Москович І.В., Скоб Ю.О., Пічугіна О.С. Основи прорамування мовою С++. Вступ до ООП. : методичний посібник – Харків : Нац. аерокосм. ун-т ім. М. Є. Жуковського "Харків. авіац. ін-т", 2024. – 124с.

3. Соколов А.Ю., Угрюмов М.Л., Халтурин В.А., Чернышев Ю.К.. Информатика [Текст] : конспект лекцій – Харьков : Нац. аерокосм. ун-т «Харк. авіац. ін-т», 2003. – 191 с.

4. Соколов, О.Ю. Інформатика для інженерів [Текст] / О.Ю. Соколов, І.Т. Зарецька, Г.М. Жолткевич, О.В. Ярова. – Харків : Факт, 2005. – 423с.

5. Юрченко І.В., Сікора В.С. Інформатика та програмування. Частина 2.– Чернівці: Видавець Яворський С.Н., 2015.– 210 с.

6. Вступ до програмування мовою С++. Організація обчислень / Ю. А. Белов, Т. О. Карнаух, Ю. В. Коваль, А. Б. Ставровський. – К.: ВПЦ “Київський університет”, 2012. – 175 с.

Допоміжна

1. Караванова Т.П. Інформатика: методи побудови алгоритмів та їх аналіз. Необчислювальні алгоритми: Навч. посіб. для 9-10 кл. із поглибл. вивч. інформатики. – К.: Генеза. – 2007.- 216 с.: іл.

2. Караванова Т.П. Інформатика: методи побудови алгоритмів та їх аналіз. Обчислювальні алгоритми: Навч. посіб. для 9-10 кл. із поглибл. вивч. інформатики – К.: Генеза. – 2008.- 333 с.: іл.

12. Інформаційні ресурси

1. https://study.korobchinskiy.com – Електронний курс курс “Основи програмування (Мова С++) / Програмування та алгоритмічні мови”
2. https://mentor.khai.edu – Електронний курс Національного аерокосмічного університету ім. М.Є. Жуковського «ХАІ» курс “Основи програмування (Мова С++)”
3. http://programming.in.ua/- Програмування українською.
4. Сайт науково-технічної бібліотеки університету. – Режим доступу: https://library.khai.edu.