

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 ПРИЗВИЩЕ)

« 21 » 08 2025 р.

**СИЛАБУС ОBOB'ЯЗKОВОЇ
НАВЧАЛЬНОЇ ДИСЦИПЛІНИ**

НАВЧАЛЬНА ПРАКТИКА

(назва навчальної дисципліни)

Галузь знань: Ф «Інформаційні технології»

(шифр і найменування галузі знань)

Спеціальність: Ф3 «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
(науковий ступінь і вчене звання)


(підпис)

Олексій КАРТАШОВ
(ім'я та ПРІЗВИЩЕ)

Погоджено з представником здобувачів освіти:

Здобувач першого (бакалаврського)

рівня вищої освіти за ОП

«Інтелектуальні системи та технології»


(підпис)

Катерина ПИЛИПЕНКО
(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; Базы 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, Навчальна практика*

Напрями наукових досліджень:

Програмування. Базы даних. III

Контактна інформація:

k.korobchinskiy@khai.edu

2. Опис навчальної дисципліни

Форма здобуття освіти	Денна
Семестр	2
Мова викладання	Українська
Тип дисципліни	Обов'язкова
Обсяг дисципліни: кредити ЄКТС/ кількість годин	3 кредитів ЄКТС / 90 годин (СРЗ – 90);
Види навчальної діяльності	Самостійна робота
Види контролю	Поточний контроль, семестровий контроль – залік
Пререквізити	“Організація та обробка електронної інформації”, “Іноземна мова”, “Основи програмування (C++)”, “Математична логіка”, “Математичний аналіз”

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

Мета вивчення: формування системи теоретичних знань та практичних компетенцій щодо методології алгоритмізації, проектування архітектури програмних додатків, а також опанування техніки застосування базових алгоритмічних структур та динамічних структур даних засобами мови програмування C#.

Для досягнення поставленої мети визначено такі **основні завдання:**

- деталізація етапів життєвого циклу розробки програмного забезпечення та імплементація принципів процедурної парадигми програмування під час проектування систем мовою C#;
- аналітичне дослідження та розробка алгоритмічних моделей для розв'язання типових економіко-математичних задач;
- експертний аналіз та освоєння функціональних можливостей сучасних інтегрованих середовищ розробки (IDE), зокрема екосистеми Microsoft Visual Studio;
- вивчення системних концепцій та архітектурних принципів функціонування програмних комплексів у середовищі ОС Windows.

Предметом навчальної дисципліни є теоретико-методологічні засади та практичний інструментарій застосування базових алгоритмічних конструкцій і структур даних у контексті сучасних технологій інженерії програмного забезпечення.

Для виконання навчальної практики потрібно опанувати й закріпити дисципліну "Основи програмування мовою C++" яка є фундаментальним складником професійної підготовки бакалаврів галузі знань "Комп'ютерні науки".

Вхідні вимоги (пререквізити): для успішної опанування навчальної практики необхідно наявність базової підготовки в обсязі загальноосвітньої програми, а також системне засвоєння знань із суміжних дисциплін: "Організація та обробка електронної інформації", "Основи програмування (C++)".

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 Знання та розуміння предметної області та розуміння професійної діяльності

ЗК6 – Здатність вчитися й оволодівати сучасними знаннями

ЗК7 – Здатність до пошуку, оброблення та аналізу інформації з різних джерел

СК3.Здатність до логічного мислення, побудови логічних висновків, використання формальних мов і моделей алгоритмічних обчислень, проектування, розроблення й аналізу алгоритмів, оцінювання їх ефективності та складності, розв'язності та нерозв'язності алгоритмічних проблем для

адекватного моделювання предметних областей і створення програмних та інформаційних систем.

СК8.Здатність проектувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: узагальненого, об'єктно-орієнтованого, функціонального, логічного, з відповідними моделями, методами й алгоритмами обчислень, структурами даних і механізмами управління.

СК17.Здатність розробляти інтелектуальне програмне забезпечення, використовувати сучасні методи штучного інтелекту.

Програмні результати навчання:

ПРН1. Застосовувати знання основних форм і законів абстрактно-логічного мислення, основ методології наукового пізнання, форм і методів вилучення, аналізу, обробки та синтезу інформації в предметній області комп'ютерних наук.

ПРН11. Володіти навичками управління життєвим циклом програмного забезпечення, продуктів і сервісів інформаційних технологій відповідно до вимог і обмежень замовника, вміти розробляти проєктну документацію (техніко-економічне обґрунтування, технічне завдання, бізнес-план, угоду, договір, контракт).

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 "Організація та обробка електронної інформації", "Іноземна мова", "Основи програмування (C++)", "Математична логіка", "Математичний аналіз".

У результаті вивчення навчальної дисципліни студенти повинні:

знати:

- Життєвий цикл ПЗ: основні етапи проектування, розгортання та супроводу програмних систем.
- Алгоритмізацію: типові керуючі конструкції (розгалуження, цикли, вибір) та методи їх оптимізації.
- Парадигми програмування: принципи процедурного, структурованого та основи об'єктно-орієнтованого підходу.
- Екосистему .NET: архітектуру керованого коду (Managed Code), принципи роботи CLR (Common Language Runtime) та JIT-компіляції.
- Систему типів (CTS): * базові типи (значущі та посилальні);
- Складні структури даних: масиви (одновимірні та багатовимірні), переліки (enum), структури (struct).
- Синтаксис та оператори: оператори управління потоком виконання, обробку винятків (try-catch-finally).
- Модульність: правила визначення та виклику методів, передачу параметрів (за значенням та за посиланням ref/out).

- Метaproгpaмyвaння: директиви препроцесора та засоби умовної компіляції.
- Потоки введення-виведення: роботу з простором назв System.IO, серіалізацію та десеріалізацію даних.
- Узагальнення (Generics): принципи створення універсальних шаблонів класів та методів для забезпечення типобезпеки.
- GUI-проекткування: принципи побудови подійно-орієнтованих додатків для ОС Windows (Windows Forms або WPF).

вміти: складати програми мовою C#, забезпечуючи:

- Алгоритмічну реалізацію: розробляти програмні модулі для обчислення складних математичних задач (курсу вищої математики) з використанням бібліотеки System.Math.
- Роботу зі структурами даних: проектувати та обробляти масиви об'єктів, здійснювати пошук, фільтрацію та сортування даних.
- Маніпуляцію даними: реалізовувати механізми читання/запису даних у файли різних форматів (текстові, бінарні).
- Інженерію інтерфейсу: створювати графічні інтерфейси користувача (GUI), використовуючи стандартні елементи управління (форми, кнопки, текстові поля, списки).
- Професійну розробку: ефективно використовувати можливості IDE (Visual Studio / JetBrains Rider) для написання коду.
- Верифікацію та супровід: проводити відладку (debugging) коду, використовувати точки зупинки (breakpoints), здійснювати рефакторинг та аналіз стеку викликів для усунення помилок.

мати уявлення:

- Об'єктно-орієнтований аналіз (OOA): фундаментальні принципи ООП (інкапсуляція, успадкування, поліморфізм, абстракція).
- Архітектурні патерни: сучасні підходи до побудови багат шарових додатків та розділення бізнес-логіки й інтерфейсу.
- Тренди розробки на C#: розуміння сучасних можливостей мови (LINQ, асинхронне програмування async/await, використання NuGet-пакетів).

4. Програма 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 представлення даних

- Архітектура ЕОМ: Функціональна схема комп'ютера: процесор, рівні кеш-пам'яті, оперативна пам'ять та периферія.
- Керування пам'яттю: Фізична та логічна адресація. Розподіл пам'яті в середовищі виконання: статична пам'ять, стек (Stack) для значущих типів та керована купа (Managed Heap) для посилальних типів.

- Організація даних: Бінарне представлення інформації. Одиниці виміру: біти, байти, машинні слова. Порядок байтів у пам'яті.
- Системне оточення: Склад та призначення сучасного ПЗ. Роль ОС та віртуальних машин. Архітектура .NET Framework / .NET Core: роль CLR (Common Language Runtime).

Тема 2. Життєвий цикл та етапи виконання програм

- Файлова організація: Структура вихідного коду та ресурсів додатка.
- Процес збірки: Етапи трансляції коду в проміжну мову (MSIL/CIL). Поняття метаданих та маніфесту збірки.
- Виконання: Робота JIT-компілятора (Just-In-Time). Типи вихідних модулів: динамічні бібліотеки (.dll) та виконувані файли (.exe).

Змістовий модуль 2. Основи алгоритмізації та методології розробки

Тема 3. Теорія алгоритмів та інженерія ПЗ

- Алгоритмізація: Формальне визначення та властивості алгоритму. Базові керуючі структури: послідовність, розгалуження, ітерація. Методи візуалізації: блок-схеми та UML Activity Diagrams.
- Парадигми програмування: Еволюція від процедурного та структурованого до об'єктно-орієнтованого підходу.
- Середовище розробки: Огляд Visual Studio. Структура проєкту та рішення (Solution). Інструменти автоматизації збірки, дебаггер (Debugger) та засоби аналізу коду.

Тема 4. Алгоритмічна декомпозиція

- Проектування: Метод «зверху-вниз» (покрокове уточнення). Оцінка складності алгоритмів (Big O notation). Типові обчислювальні задачі та їх алгоритмічна реалізація.

Змістовий модуль 3. Базовий синтаксис та система типів C#

Тема 5. Лексичний склад та лінійні програми

- Алфавіт та синтаксичні одиниці: Ключові слова, ідентифікатори, константи та коментарі. Стандарти іменування (PascalCase, camelCase).
- Змінні та вирази: Оголошення, ініціалізація та область видимості. Побудова арифметичних та логічних виразів.

Тема 6. Система типів даних (CTS)

- Класифікація: Значущі типи (Value Types) та посилальні типи (Reference Types).
- Базові типи: Цілі числа, числа з плаваючою комою (double, decimal), логічний тип (bool), символи (char).
- Перетворення типів: Неявне розширення та явне приведення (casting). Безпечне перетворення за допомогою операторів is, as та класу Convert.
- Операції: Унарні, бінарні та тернарні оператори. Порозрядні (bitwise) операції. Таблиця пріоритетів та асоціативність.

Тема 7. Стандартні бібліотеки

- Математичні обчислення: Використання статичного класу System.Math.
- Константи: Визначення констант (const) та полів тільки для читання (readonly). Робота з переліками (enum).

Змістовий модуль 4. Управління потоком виконання та складні типи даних

Тема 8. Оператори управління та розгалуження

- Керуючі конструкції: Умовні оператори if-else, оператор множинного вибору switch (включаючи switch-вирази).
- Робота з масивами: Оголошення та ініціалізація одновимірних та багатовимірних масивів. Алгоритми обходу, фільтрації та пошуку елементів.

Тема 9. Циклічні процеси

- Типи циклів: while, do-while, ітераційний цикл for, перебір колекцій foreach.
- Управління ітераціями: Оператори break та continue. Вкладені цикли та стратегії їх оптимізації.

Тема 10. Відладка та обробка помилок

- Налагодження: Типи помилок (синтаксичні, логічні, помилки виконання). Кроковий режим (Step Into/Over).
- Виняткові ситуації: Блоки try-catch-finally. Створення та генерування власних винятків.

Змістовий модуль 5. Обробка посилальних типів та робота з ресурсами

Тема 11. Рядкова інформація та символи

- Рядки в C#: Клас System.String. Імутабельність (незмінність) рядків. Ефективна побудова рядків через StringBuilder. Операції пошуку, форматування та регулярні вирази.

Тема 12. Робота з файловою системою

- Введення-виведення (I/O): Простір назв System.IO. Текстові та бінарні файли. Потоки даних (Streams). Класи File, Directory, Path.

Тема 13. Управління об'єктами у пам'яті

- Посилання та вказівники: Відмінність між посиланням та фізичною адресою. Робота з купою.
- Garbage Collection (GC): Принципи автоматичного звільнення пам'яті. Некеровані ресурси та патерн Dispose.

Змістовий модуль 6. Модульність та процедурне програмування в C#

Тема 14. Методи та параметри

- Проектування методів: Оголошення, параметризація та повернення значень. Статичні методи.
- Передача даних: Параметри за значенням та за посиланням (модифікатори ref, out, in). Необов'язкові та іменовані аргументи.
- Перевантаження методів: Поліморфізм на рівні сигнатур.

Тема 15. Узагальнення (Generics)

- Шаблони: Поняття узагальненого програмування. Узагальнені методи та класи.
- Колекції: Робота з List<T>, Dictionary<TKey, TValue>. Переваги над нетипізованими масивами.

Змістовий модуль 7. Об'єктно-орієнтоване програмування (ООП)

Тема 16. Класи та структури

- Типи користувача: Оголошення класів. Поля, властивості (Properties), методи. Конструктори та їх перевантаження.
- Encapsulation: Рівні доступу (public, private, protected, internal).

Тема 17. Спадкування та Поліморфізм

- Ієрархія класів: Базові та похідні класи. Перевизначення методів (virtual, override). Абстрактні класи та інтерфейси.
- Зв'язки між об'єктами: Асоціація, агрегація та композиція.

Тема 18. Проектування систем (UML та Патерни)

- Візуальне моделювання: Діаграми класів (Class Diagrams). Відображення структури додатка та взаємодії об'єктів.
- Основи архітектури: Огляд шаблонів проектування (Design Patterns) та їх роль у створенні масштабованих систем.

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, заліку).

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. Практичні роботи.
3. Залікова роботи.

Форма підсумкового контролю успішності навчання: **залік**.

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

Таблиця 7.1 – Розподіл балів, які отримують здобувачі освіти

Поточне тестування та самостійна робота										Сума	Підсумковий тест (залік)*
Модуль №1											
T1	T2	T3	T4	T5	T6	T7	T8	T9	T10		
10	10	10	10	10	10	10	10	10	10	100	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 2 *теоретичних та 2 практичних питань та максимальна кількість балів за кожне питання 25 балів (сума – 100 балів)*.

7.2. Якісні критерії оцінювання

Необхідний обсяг знань для одержання позитивної оцінки:

- Життєвий цикл ПЗ: основні етапи проектування, розробки та супроводу програмних продуктів.
- Алгоритмізація: типові алгоритмічні конструкції (лінійні, розгалужені, циклічні) та методи їх формалізації.
- Парадигми програмування: принципи процедурного, структурованого та базові поняття об'єктно-орієнтованого програмування.
- Архітектура .NET: принципи роботи керованого коду, роль середовища виконання CLR та системи типів CTS.
- Інструментарій розробки: особливості використання інтегрованого середовища (IDE) Visual Studio для написання та відладки програм.
- Система типів C#: базові типи даних (цілочисельні, дійсні, логічні, символьні) та специфіка їх зберігання у пам'яті (Stack vs Heap).
- Складні структури даних: правила роботи з масивами (одновимірними та багатовимірними), рядками (string), структурами (struct) та переліками (enum).
- Керуючі конструкції: оператори вибору (if, switch) та оператори повторення (for, while, do-while, foreach).

- Методи: правила оголошення та виклику методів, способи передачі параметрів (значення, посилання ref, вихідні параметри out).
- Система введення-виведення: робота з консоллю та основи використання простору назв System.IO для взаємодії з файловою системою.
- Виняткові ситуації: базові принципи обробки помилок за допомогою блоків try-catch.

Необхідний обсяг вмінь для одержання позитивної оцінки:

- Розробка програмного коду: складати програмні додатки мовою C# для автоматизації обчислювальних процесів.
- Математичне моделювання: реалізовувати алгоритми для розв'язання задач з курсу вищої математики та економіко-математичного моделювання.
- Робота з даними: створювати та обробляти масиви об'єктів/структур (пошук мінімального/максимального, сортування, фільтрація).
- Файлові операції: реалізовувати збереження результатів обчислень у текстові файли та зчитування даних із них.
- Структуризація коду: використовувати методи для декомпозиції складних задач на простіші підзадачі.
- Робота з інтерфейсом: проектувати базову взаємодію з користувачем через консольний інтерфейс або прості графічні форми.
- Відладка та тестування: самостійно виявляти синтаксичні та логічні помилки, використовувати точки зупинки (breakpoints) та покрокове виконання програми для аналізу стану змінних.
- Супровід коду: вносити зміни до існуючих програм, проводити рефакторинг та забезпечувати читабельність коду згідно з угодами про іменування (C# Coding Conventions).

7.3 Критерії оцінювання роботи студента протягом семестру

Оцінка Задовільно (60-74 бали) виставляється студенту, який продемонстрував необхідний мінімум теоретичних знань та практичних навичок, що передбачає успішний захист усіх індивідуальних завдань та позитивний результат тестування. Студент повинен орієнтуватися в основних етапах життєвого циклу проектування програмного забезпечення, розуміти логіку функціонування типових алгоритмічних конструкцій та володіти базовими принципами процедурного і структурованого програмування. Практична підготовка має включати вміння розробляти додатки мовою C# для вирішення типових обчислювальних задач вищої математики, створювати та виконувати базову обробку значущих типів даних, структур та масивів, а також реалізовувати найпростіші операції з файловою системою.

Оцінка Добре (75-89 балів) передбачає впевнене володіння навчальним матеріалом, своєчасний захист усіх практичних робіт, виконання контрольних заходів та завдань для позааудиторної самостійної роботи. Студент має вільно оперувати складними типами даних, такими як переліки, багатовимірні масиви та посилальні типи. Необхідно демонструвати навички ефективного використання операторів управління програмою, володіти методологією створення методів із різними способами передачі параметрів та розуміти принципи організації пам'яті. Рівень практичних вмінь повинен дозволяти здійснювати комплексну обробку файлових потоків, проектувати прості класи та розробляти базові елементи графічного інтерфейсу користувача.

Оцінка Відмінно (90-100 балів) свідчить про системне та глибоке засвоєння всіх розділів дисципліни. Студент має скласти всі контрольні точки з оцінкою «відмінно», демонструючи здатність до творчого застосування отриманих знань. Це передбачає досконале володіння об'єктно-орієнтованою парадигмою, включаючи практичну реалізацію принципів інкапсуляції, успадкування та поліморфізму. Студент повинен вміти проектувати гнучку архітектуру програмного забезпечення на основі інтерфейсів та узагальнень, проводити професійну відладку та профілювання коду, а також використовувати засоби візуального моделювання для опису структури системи.

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:

- тематики індивідуальних завдань;
- питання, тести для контрольних заходів;
- каталоги інформаційних ресурсів.

Увесь науково методичний комплект з дисципліни розміщено на освітньому порталі: <https://stm.korobchinskiy.com/course/view.php?id=60>

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 програмування мовою C++. 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. Информатика 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. Информатика та програмування. Частина 2.– Чернівці: Видавець Яворський С.Н., 2015.– 210 с.
6. Вступ до програмування мовою C++. Організація обчислень / Ю. А. Белов, Т. О. Карнаух, Ю. В. Коваль, А. Б. Ставровський. – К.: ВПЦ “Київський університет”, 2012. – 175 с.
7. Закон України «Про вищу освіту», Указ президента України від 04.07.2005р. №1013/2005 «Про невідкладні заходи щодо забезпечення функціонування та розвитку освіти в Україні»

Допоміжна

1. Караванова Т.П. Информатика: 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.

3. Положення про проведення практики студентів вищих навчальних закладів України. URL: <https://zakon.rada.gov.ua/laws/show/z0035-93#Text>

12. Інформаційні ресурси

1. <https://study.korobchinskiy.com> – Електронний курс курс “Основи програмування (Мова C++) / Програмування та алгоритмічні мови”

2. <https://mentor.khai.edu> – Електронний курс Національного аерокосмічного університету ім. М.Є. Жуковського «ХАІ» курс “Основи програмування (Мова C++)”

3. <http://programming.in.ua/> - Програмування українською.

4. Сайт науково-технічної бібліотеки університету. – Режим доступу: <https://library.khai.edu>.