



# Алгоритмізація та програмування

## Робоча програма навчальної дисципліни (Силабус)

### Реквізити навчальної дисципліни

Рівень вищої освіти	Перший (бакалаврський)
Галузь знань	12 Інформаційні технології
Спеціальність	122 Комп'ютерні науки
Освітня програма	Системи і методи штучного інтелекту Інтелектуальні сервіс-орієнтовані розподілені обчислювання
Статус дисципліни	Нормативна
Форма навчання	очна(денна)/змішана
Рік підготовки, семестр	1 курс, осінній та весняний семестр
Обсяг дисципліни	210 годин/7 кредитів ECTS Всього аудиторних годин – 126, з них 72 години – лекції, 54 години – лабораторні заняття. Самостійна робота студентів – 84 години
Семестровий контроль/ контрольні заходи	Заліки / Модульні контрольні роботи
Розклад занять	Для груп ДА-21-23: лекції (середа), лабораторні заняття (середа) Для груп КІ-21-21: лекції (четвер), лабораторні заняття (четвер)
Мова викладання	Українська
Інформація про керівника курсу / викладачів	Лектор: Романов Валерій Володимирович, v.romanov2516@gmail.com Лабораторні: к.т.н. Безносик Олександр Юрійович, beznosyk.oleksandr@lll.kpi.ua к.е.н., доцент Просянкін-Жарова Тетяна Іванівна, t.pruman@gmail.com
Розміщення курсу	<a href="https://do.ipu.kpi.ua/course/view.php?id=393">https://do.ipu.kpi.ua/course/view.php?id=393</a>

### Програма навчальної дисципліни

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

Дисципліна "Алгоритмізація та програмування" спрямована на ознайомлення студентів із сучасними науковими концепціями, поняттями, методами алгоритмізації та технологій програмування, властивостями і особливостями алгоритмів, а також процесів створення працездатних програм мовою високого рівня. Завданням дисципліни є вивчення типових алгоритмічних конструкцій та засобів представлення алгоритму; отримання знання про синтаксис і семантику базових конструктивних елементів мови програмування: лексем, виразів та операторів; отримання знання про різновиди типів даних, як простих, так і складених (масивів, структур, об'єднань); усвідомлення особливості програмування функцій; усвідомлення парадигми імперативного програмування; вивчення особливостей програмування динамічних та файлових структур даних; отримання знання про основні прийоми структурного програмування; формування комплексного уявлення про етапи розробки програми, основні поняття та методи технологій програмування; оволодіння прийомами та технологією налагодження та тестування програм; отримання знання про основні вимоги до документування програмних продуктів.

**Мета:** ознайомлення студентів із мовами програмування для вирішення прикладних задач, сучасними поглядами на алгоритмічні процеси; набуття студентами навичок розробки програм на мові C/C++, застосування математичного апарату для створення комп'ютерних програм.

**Предмет вивчення:** теоретичні основи алгоритмізації та програмування, техніка застосування у програмуванні базових алгоритмічних структур і базових структур даних, практичні навички професійного володіння комп'ютером та процедурно-орієнтована технологія розробки програмних продуктів під час вирішення прикладних завдань в галузі комп'ютерних наук.

**Загальні компетентності:**

**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.

**ЗК 6** Здатність вчитися і оволодівати сучасними знаннями.

**Фахові компетентності:**

**ФК 3** 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.

**ФК 8** 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.

**Програмні результати навчання:**

**ПРН 1** 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.

**ПРН 5** Проектувати, розробляти та аналізувати алгоритми розв'язання обчислювальних та логічних задач, оцінювати ефективність та складність алгоритмів на основі застосування формальних моделей алгоритмів та обчислюваних функцій.

**ПРН 9** Розробляти програмні моделі предметних середовищ, вибирати парадигму програмування з позицій зручності та якості застосування для реалізації методів та алгоритмів розв'язання задач в галузі комп'ютерних наук.

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

Дисципліна «Алгоритмізація та програмування» вивчається у 1 та 2 семестрах, тому **для успішного засвоєння дисципліни** необхідні знання шкільного курсу математики та інформатики на рівні, вищому за «базовий» (для першого семестру) та дискретна математика, математичний аналіз, аналітична геометрія та лінійна алгебра, алгоритми і структури даних (для другого семестру).

**На результатах навчання з дисципліни «Алгоритмізація та програмування» базуються такі дисципліни**, як «Проектування та аналіз обчислювальних алгоритмів», «Об'єктно-орієнтоване програмування», «Технології створення програмних сервісів».

## **3. Зміст навчальної дисципліни**

**Кредитний модуль 1. Базові концепції програмування.**

**Тема 1. Введення в курс. Цілі і завдання дисципліни, її структура, зміст. Інструментальне програмне забезпечення.**

**Зміст.** Сутність інформаційних технологій. Історія мов програмування. Огляд сучасних мов програмування, рейтинг. Класифікація і вибір мов програмування. Інформаційні комп'ютерні терміни. Етапи вирішення завдань на комп'ютері. Засоби запису алгоритмів. Позначення, які використовуються у блок-схемах.

**Тема 2. Технологія програмування на мові C. Компілятори і IDE (Інтегровані середовища розробки програм).**

**Зміст.** Системи програмування. Етапи перетворення вихідного модуля у виконуваний модуль. Спрощена модель компілятора. Лексичний, синтаксичний аналізатори, генератор коду, таблиці. Компілятори та інтегровані оболонки для мови C. Середовища Dev C++, Visual Studio, CodeBlocks. Встановлення IDE.

**Тема 3. Введення в розробку програмного забезпечення. Основи алгоритмізації обчислювальних процесів.**

**Зміст.** Етапи вирішення завдань на комп'ютері. Структурне програмування. Програмування "знизу-вгору". Програмування "згори-донизу". Цілі структурного програмування. Принципи структурного програмування. Процес розробки програмного забезпечення. Основи алгоритмізації обчислювальних процесів.

**Тема 4. Структура програми на мові C. Синтаксис мови. Типи даних.**

**Зміст.** Структура програми. Стандартна бібліотека мови C. Синтаксис мови. Алфавіт, ідентифікатори, символи-роздільники, спеціальні символи. Константи. Ціла, десяткова, восьмерична, шістнадцятирична константи, константи з плаваючою крапкою, символні константи, перевизначені константи. Символьний рядок. Ключові слова. Коментарі. Введення довгих, беззнакових чисел і чисел з плаваючою крапкою. Базові типи даних за розміром пам'яті і областю значень. Перетворення типів.

**Тема 5. Вирази, операнди, операції. Математичні функції мови C/C++.**

**Зміст.** Вирази, операнди, операції. Таблиця пріоритетів операцій. Унарні, бінарні, тернарні операції. Адитивні, мультиплікативні операції. Операції зсуву, відношення. Логічні, порозрядні логічні операції. Основні математичні функції C.

**Тема 6. Основні оператори мови C. Програмування базових алгоритмічних конструкцій.**

**Зміст.** Оператори управління в C. Умовний оператор, приклади використання оператора if ... else. Оператори циклу і операція кома. Приклад використання операції кома в операторі for. Порівняння виконання операторів while і do ... while. Особливості трьох типів циклу, приклад. Оператори вибору, переходу. Оператор break і оператор продовження continue. Приклади використання операторів циклу і умовного оператора. Схеми виконання в циклах операторів break і continue.

**Тема 7. Введення-виведення даних в C.**

**Зміст.** Форматне виведення даних в C. Приклади виведення з використанням функції printf(). Основні символи типів в специфікаціях формату мови C. Приклади. Виведення цілих значень по формату %d. Виведення дійсних значень по форматам %f і %e. Виведення рядка символів по формату %s. Форматне введення даних в C, функція scanf(). Модифікатори форматів введення та виведення. Приклади завдання специфікацій формату. Введення-виведення окремих символів. Введення-виведення рядків.

**Тема 8. Масиви.**

**Зміст.** Загальні відомості. Приклади оголошення і використання масивів. Вказівники, їхні властивості, призначення, використання. Використання вказівників у роботі з масивами. Створення вказівника на масив. Алгоритми обробки масивів в C: введення-виведення, зміна даних в масиві, множення масивів. Алгоритми обробки матриць.

**Тема 9. Функції.**

**Зміст.** Загальний формат, визначення, оголошення – прототип функції, виклик функції, виклик по значенню, виклик з посиланням, опис типів функції. Передача параметрів, повернення з

функції, повернення значень функції. Рівноправність функцій. Вказівники в параметрах функцій. Масиви і рядки, як параметри функцій. Функції із змінною кількістю параметрів. Аргументи функцій, аргументи функції main().

### **Тема 10. Вказівники.**

**Зміст.** Загальні положення. Приклади оголошення вказівників. Використання специфікатора const з вказівниками. Операції над вказівниками. Вказівники на функції. Масиви вказівників. Складні описувачі типів. Функції зі змінним числом аргументів.

### **Тема 11. Структури даних в програмуванні.**

**Зміст.** Рівні представлення структур даних. Класифікація структур даних. СД типу вектор. СД типу масив. "Вільні" масиви. Трикутні матриці. Розріджені матриці. СД типу запис-структура. Оголошення структур і визначення структурних об'єктів. Ініціалізація і доступ до даних структури. Присвоювання структур. Масиви структур. Структури з посиланнями на себе. Визначення розміру пам'яті, що виділяється під структуру. Відмінність оголошень структур в С і в С++. Вказівники на структури.

### **Тема 12. Динамічні структури даних. Стек і Купа.**

**Зміст.** Загальні положення. СД типу однозв'язного лінійного списку. Структури з бітовими полями. СД типу таблиця. СД типу хеш-таблиця. Стеки. Черги. Двозв'язні списки. Бінарні дерева. Стек як структура даних. Сегменти, сегменти коду, сегменти даних. Купа. Стек викликів. Переповнення стеку.

### **Тема 13. Класи пам'яті.**

**Зміст.** Автоматичні змінні. Зовнішні змінні. Зовнішні статичні змінні. Внутрішні статичні змінні. Реєстрові змінні.

### **Тема 14. Робота з пам'яттю. Псевдовипадкові числа. Генерація випадкових чисел.**

**Зміст.** Робота з пам'яттю. Способи виділення пам'яті. Адресація і представлення в пам'яті. Типове уявлення віртуальної пам'яті процесу в операційних системах. Динамічний розподіл пам'яті в С, приклади. Динамічне виділення і звільнення пам'яті в С/С++ за допомогою операцій new та delete. Виділення і звільнення пам'яті під двомірний динамічний масив. Приклади роботи з динамічними масивами. Псевдовипадкові числа. Генерація випадкових чисел. Функції rand() і srand(). Отримання цілих випадкових чисел в заданих діапазонах. Отримання дійсних випадкових чисел. Рівноймовірні випадкові числа.

### **Тема 15. Рекурсія і рекурсивні функції.**

**Зміст.** Загальні відомості. Приклади простих рекурсивних функцій. Приклад обчислення виразу з врахуванням рекурентних залежностей. Різниця між ітерацією і рекурсією. Методи рекурсивного програмування. Форми рекурсивного звернення. Форма виконання дій до рекурсивного виклику. Форма виконання дій після рекурсивного виклику. Форма виконання дій як до, так і після рекурсивного виклику. Виконання дій на рекурсивному спуску. Виконання дій на рекурсивному поверненні. Виконання дій на рекурсивному спуску і поверненні.

### **Тема 16. Найпростіші алгоритми в програмуванні.**

**Зміст.** Цілочисельні алгоритми. Алгоритм Евкліда. Словесна і блочна схеми представлення алгоритму. Операторна схема і псевдокод для представлення алгоритму. Алгоритм знаходження найменшого спільного кратного і приклад використання алгоритму Евкліда. Алгоритм одержання простих чисел. Алгоритм розкладу чисел на прості множники. Алгоритми перетворення ланцюгового дроби у десяткове число і навпаки. Алгоритм отримання N-го числа Фібоначі. Ділення числа А на число В з довільною точністю. Алгоритми переходу від однієї системи числення до іншої. Числові послідовності. Обчислювальні операції з поліномами. Алгоритм послідовних приближень. Метод Герона. Чисельне рішення рівнянь. Методи дихотомії, хорд, Ньютона. Прості алгоритми

пошуку. Лінійний пошук. Бінарний пошук. Алгоритми сортування.

### **Тема 17. Символи та рядки. Обробка символів та рядків в C.**

**Зміст.** Засоби мови C для обробки символної інформації. Введення-виведення рядків. Функції для роботи з рядками символів. Масиви рядків. Конкатенація рядків. Пошук символів та підрядків у рядку. Перетворення рядків у числа. Операції із блоками пам'яті.

### **Тема 18. Робота з датою та часом в C.**

Елементи структури tm. Функції стандартної бібліотеки C для операцій з календарним і розділеним часом. Специфікатори формату виведення дат і часу в C. Приклади роботи з датами.

### **Тема 19. Робота з файлами в C.**

**Зміст.** Організація роботи з файлами в C. Функції введення-виведення нижнього і верхнього рівнів. Операції над файлами. Допустимі значення режимів доступу до файлу. Приклади використання функцій fopen і freopen. Введення/виведення форматуваних даних у файлах. Основні операції роботи з файлами: заповнення файлу даними, виведення даних для перегляду, копіювання вмісту файлу, пошук поля або полів, що відповідають заданому критерію; додавання записів в файл; зміна значень полів у записі; видалення записів з файлу. Файлове введення-виведення символів і рядків. Файлове введення/виведення неформатованих даних. Приклад використання функцій getc() і ungetc(). Функції fputs() і fgets(), приклад використання функцій. Функції fwrite() і fread(), приклад використання функцій. Функції управління доступом к даним у файлі. Функції fseek(), ftell() і rewind(). Точки відліку SEEK\_SET, SEEK\_CUR, SEEK\_END. Використання функції fseek() для коригування даних в двійковому файлі з використанням довільного доступу до даних. Використання функцій fseek() і rewind() для видалення запису з файлу.

### **Тема 20. Препроцесор C. Директиви препроцесора.**

**Зміст.** Директиви препроцесора. Вставка іменованих констант і макровизначень; включення файлів; умовна компіляція; управління нумерацією рядків; обробка помилок. Директива #define, використання директиви #define для визначення іменованих констант. Використання директиви #define для макровизначення. Включення файлів. Директива #include, використання оператора #include. Використання операції "##". Умовна компіляція. Операція defined ідентифікатор. Використання директив умовної компіляції. Директиви #ifdef ідентифікатор. Псевдозмінні, управління нумерацією рядків і директива обробки помилок. Директива #line. Використання директиви #error.

### **Тема 21. Використання графічних можливостей C/C++.**

**Зміст.** Загальні положення. Введення в консольне вікно Windows. Короткий огляд графічних бібліотек C++. Робота з простим графічним редактором. Способи представлення функцій. Системи координат і позначення. Побудова графіка. Приклади графічного переміщення об'єктів. Анімація.

### **Тема 22. Використання графічних можливостей C/C++. Основи OpenGL.**

**Зміст.** Основні можливості. Інтерфейс OpenGL. Архітектура OpenGL. Підключення бібліотеки OpenGL. Завантаження і установка GLUT для Visual Studio 2013. Завантаження і установка GLUT для Dev C++. Установка GLUT для Ubuntu. Ознайомлення з основами роботи GLUT. Перетворення простору в OpenGL. Примітиви символів в OpenGL. Опис примітивів. Геометричні примітиви OpenGL. Функції зворотного зв'язку. Інтерактивне введення. Таймер в OpenGL. Приклади використання OpenGL. Відскакуючий квадрат. Обертання прямокутника навколо точки. Переміщення квадрату по екрану за допомогою клавіш. Дослідження області визначення і побудова графіка функції. Зразок виведення результатів. Література по OpenGL.



### **Тема 23. Використання графічних можливостей C/C++. Програмування в консольному вікні.**

**Зміст.** Елементи програмування в консольному вікні. Управління консольним вікном в «текстовому режимі». Малювання в консольному вікні. Графічні примітиви, інструменти малювання. Малювання точки, лінії. Поточна позиція пера. Вибір пера. Вибір режиму фону. Вибір режиму малювання. Вибір пензля. Дослідження області визначення і побудова графіка функції.

### **Тема 24. Підсумки 1-го семестру. Контрольна робота 1.**

#### **Кредитний модуль 2. Процедурне програмування.**

### **Тема 25. Робота з довгими числами.**

**Зміст.** Представлення довгих чисел, Перетворення і обробка довгих чисел. Операції додавання, віднімання, множення та ділення довгих чисел. Алгоритм множення Карацуби. Алгоритм множення Тоома.

### **Тема 26. Обробка символів і рядків в C++. Операції з блоками пам'яті.**

**Зміст.** Обробка рядків в C++. Створення, присвоювання, конкатенація рядків. Звернення по індексам к елементам рядків. Порівняння рядків. Підрядки. Пошук в рядках. Операції з блоками пам'яті.

### **Тема 27. Потокове введення-виведення у C++.**

**Зміст.** Введення у стандартну бібліотеку C++. Класи IOStream, String. Стандартні потоки введення-виведення в C++. Маніпулятори потоків. Файлові потоки введення-виведення. Режими для відкриття файлу. Методи об'єктів (класів) файлових потоків. Обробка файлових записів. Прямий доступ до даних в файлі.

### **Тема 28. Багатомодульні програми в C. Бібліотеки. Створення бібліотек.**

**Зміст.** Використання багатомодульної програми і динамічної пам'яті. Передача аргументів в функцію main(). Приклад створення бібліотеки. Створення статичної бібліотеки. Створення динамічної бібліотеки. Використання бібліотеки в програмі. Компіляція програми зі статичною бібліотекою. Компіляція програми з динамічною бібліотекою.

### **Тема 29. Нові можливості нового стандарту C/C++.**

**Зміст.** Цикл for\_each, введення в std::array, std::vector в мові C++. Цикл for\_each і ключове слово auto. Цикл for\_each і посилання. Рядки std::string, рядки C-style, клас std::string\_view.

### **Тема 30. Бібліотека STL.**

**Зміст.** Навіщо потрібна STL. Модель програмування STL. Послідовності. Адаптери контейнерів. Асоціативні контейнери. Ітератори. Ітерація по елементам структур даних. Ітератори стандартної бібліотеки C++. STL – контейнери послідовностей. Клас-шаблон std::vector. Методи шаблону std::vector. Клас-шаблон std::deque. Клас-шаблон std::list. Адаптер std::stack. Адаптер std::queue. Асоціативні контейнери. Контейнер std::set. Контейнер std::multiset. Контейнер std::map.

### **Тема 31. Програми і алгоритми в Стандартній бібліотеці C++.**

**Зміст.** Бібліотека алгоритмів. Алгоритм std::find() і пошук елемента за значенням. Алгоритм std::find\_if() і пошук елемента з умовою. Алгоритми std::count()/std::count\_if і підрахунок входжень елемента. Алгоритм std::sort() і користувачьке сортування. Алгоритм std::for\_each() і всі елементи контейнера. Порядок виконання. Діапазони в C++20.

### **Тема 32. Введення у об'єктно-орієнтоване програмування. Основні концепції.**

**Зміст.** Відмінності мови C++ від C. Класи, Об'єкти і Методи класів. Робота з класами. Оголошення, елементи класів. Специфікатори доступу: видимість елементів класу – public і private. Структури і класи. Елементи-дані. Елементи-функції. Ініціалізація. Область видимості і час життя об'єктів. Вбудовані функції. Конструктори та деструктори. Конструктори з аргументами за

замовчуванням. Конструктори за замовчуванням. Перевантажені конструктори. Використання деструкторів. Елементи класів. Вказівник this. Робота з private-елементами даних. Статичні елементи. Шаблони функцій. Шаблони класів.

### Тема 33. Підсумки 2-го семестру. Контрольна робота 2.

#### 4. Навчальні матеріали та ресурси

##### Базова література

1. Трофименко О.Г. С++. Алгоритмізація та програмування : підручник / О.Г. Трофименко, Ю.В. Прокоп, Н.І. Логінова, О.В. Задерейко. 2-ге вид. перероб. і доповн. – Одеса : Фенікс, 2019.
2. Васильєв, О. Програмування на С++ в прикладах і задачах : навчальний посібник / О.Васильєв. – Київ :Видавництво Ліра-К, 2020.
3. Тверитникова О. Є. Базові алгоритми та основи програмування: теорія і практика : навчальний посібник для студентів спеціальностей "Автоматизація та комп'ютерно-інтегровані технології", "Метрологія та вимірювальна техніка" усіх форм навчання / О.Є. Тверитникова, В.А. Крилова, О.Г. Васильченков ; Міністерство освіти і науки України, Національний технічний університет "Харківський політехнічний інститут". Харків :НТУ "ХПІ" ; А.М. Панов, 2020.
4. Ковалюк Т. В. Алгоритмізація та програмування: підручник для студентів вищих навчальних закладів, що навчаються за напрямками "Комп'ютерні науки", "Комп'ютерна інженерія", "Програмна інженерія" / Т.В. Ковалюк ; Міністерство освіти і науки, молоді та спорту України. Львів :Видавництво "Магнолія 2006", 2019.
5. Козак Л. І. Основи програмування: навчальний посібник / Л.І. Козак, І.В. Костюк, С.П. Стасевич. – Львів : Новий Світ – 2000, 2019.

##### Додаткова література

6. Вступ до програмування мовою С++. Організація обчислень: навч. посіб. / Ю. А. Белов, Т. О. Карнаух, Ю. В. Коваль, А. Б. Ставровський. – К. : Видавничо-поліграфічний центр "Київський університет", 2012. – 175 с.

#### Навчальний контент

##### 5. Методика опанування навчальної дисципліни (освітнього компонента)

№ п/п	Змістові модулі / теми	Кількість годин, відведених на:			Термін виконання
		лекції	лаб. заняття	сам. робота.	
1	2	3	4	5	
<b>Кредитний модуль 1. Базові концепції програмування.</b>					
T1	Введення в курс. Цілі і завдання дисципліни, її структура, зміст. Інструментальне програмне забезпечення.	2			1-й тиждень
T2	Технологія програмування на мові С. Компілятори і IDE (Інтегровані середовища розробки програм).	4	2	4	2-й тиждень
T3	Введення в розробку програмного забезпечення. Основи алгоритмізації обчислювальних процесів.	2	2		3-й тиждень
T4	Структура програми на мові С. Синтаксис мови. Типи даних.	2	2	2	4-й тиждень
T5	Вирази, операнди, операції. Математичні функції мови С/С++.	2	2	2	4-й тиждень

T6	Основні оператори мови C. Програмування базових алгоритмічних конструкцій.	2	2	2	5-й тиждень
T7	Введення-виведення даних в C.	2	2	4	6-й тиждень
T8	Масиви.	2	8	4	6-й тиждень
T9	Функції.	2	4	4	7-й тиждень
T10	Вказівники.	2	2	4	8-й тиждень
T11	Структури даних в програмуванні.	2	4	2	8-й тиждень
T12	Динамічні структури даних. Стек і Купа.	4		4	9-й тиждень
T13	Класи пам'яті.	2		2	10-й тиждень
T14	Робота з пам'яттю. Псевдовипадкові числа. Генерація випадкових чисел.	2		2	11-й тиждень
T15	Рекурсія і рекурсивні функції.	2	4	4	12-й тиждень
T16	Найпростіші алгоритми в програмуванні.	2		4	12-й тиждень
T17	Символи та рядки. Обробка символів та рядків в C.	2		4	13-й тиждень
T18	Робота з датою та часом в C.	2			14-й тиждень
T19	Робота з файлами в C.	4		4	15-й тиждень
T20	Препроцесор C. Директиви препроцесора.	2		2	16-й тиждень
T21	Використання графічних можливостей C/C++.	2		2	16-й тиждень
T22	Використання графічних можливостей C/C++. Основи OpenGL.	2		2	17-й тиждень
T23	Використання графічних можливостей C/C++. Програмування в консольному вікні.	2		2	18-й тиждень
T24	Підсумки 1-го семестру. Контрольна робота 1.	2	2		18-й тиждень
	<b>Всього модуль 1</b>	<b>54</b>	<b>36</b>	<b>60</b>	
<b>Кредитний модуль 2. Процедурне програмування.</b>					
T25	Робота з довгими числами.	2	2	2	2-й тиждень
T26	Обробка символів і рядків в C++. Операції з блоками пам'яті.	2	2	2	4-й тиждень



T27	Потокове введення-виведення у C++.	2	4	4	6-й тиждень
T28	Багатомодульні програми в C. Бібліотеки. Створення бібліотек.	2	2	2	8-й тиждень
T29	Нові можливості нового стандарту C/C++.	2	2	2	10-й тиждень
T30	Бібліотека STL.	2	2	2	12-й тиждень
T31	Програми і алгоритми в Стандартній бібліотеці C++.	2	2	4	14-й тиждень
T32	Введення у об'єктно-орієнтоване програмування. Основні концепції.	2		6	16-й тиждень
T33	Підсумки 2-го семестру. Контрольна робота 2.	2	2		18-й тиждень
	<b>Всього модуль 2</b>	<b>18</b>	<b>18</b>	<b>24</b>	

#### Лабораторні заняття

Основні завдання циклу лабораторних занять полягають у закріпленні теоретичного матеріалу та отриманні студентами практичних навичок з розробки системного програмного забезпечення та налаштування сучасних операційних систем.

№ з/п	Назва лабораторної роботи	Кількість ауд. годин
	<b>Кредитний модуль 1. Базові концепції програмування.</b>	
1	Програмування базових алгоритмічних конструкцій	4
2	Виконання обчислень за допомогою рекурентних співвідношень	4
3	Обробка числових послідовностей. Використання циклів	4
4	Одновимірні масиви	4
5	Двовимірні масиви	4
6	Функції. Область дії. Передача змінних і масивів у функції	6
7	Рекурсія. Рекурсивні функції	4
8	Структури	4
	Залік	2
	<b>Кредитний модуль 2. Процедурне програмування.</b>	
9	Операції з довгими числами	2
10	Робота з рядками	2
11	Робота з файлами	2
12	Бінарні файли	4
13	Директиви препроцесора	2
14	Побудова графіків функцій	4
	Залік	2

#### 6. Самостійна робота студента

T1	Розв'язок задач із представлення алгоритму в вигляді блок-схем алгоритмів	5
T2	Розв'язок задач опрацювання складних арифметичних формул, табуляція функцій	5

T3	Розв'язок задач з опрацювання одномірних масивів чисел	5
T4	Розв'язок задач із опрацювання двомірних масивів чисел	5
T5	Розв'язок задач із опрацювання символів та рядків	6
T6	Розв'язок задач із опрацювання різних алгоритмів сортування засобами мови С	5
T7	Динамічна пам'ять	6
T8	Опрацювання рекурсивних функцій. Перезавантаження функцій	6
T9	Набуття навичок роботи з масивами записів	5
T10	Робота з файлами	4
T11	Використання зв'язних списків	4
T12	Обробка виключень	4
T13	Модулі, багатомодульне програмування	4
T14	Створення бібліотек, використання вбудованих бібліотек	4
T15	Потокове введення, розв'язок задач	4
T16	Робота з графікою в С++	4
T17	Введення у об'єктно-орієнтоване програмування на мові С++. Основні концепції	4
T18	Розробка підпрограм користувача (процедур і/або функцій) при проектуванні складних програмних комплексів	4
	<b>Всього</b>	<b>84</b>

## Політика та контроль

### 7. Політика навчальної дисципліни (освітнього компонента)

Вимоги, яких має дотримуватися студент в рамках даної дисципліни:

- правила відвідування занять: відвідування лекцій та лабораторних робіт, а також відсутність на них, не оцінюється. Однак, студентам рекомендується відвідувати заняття, оскільки на них викладається теоретичний матеріал та розвиваються навички, необхідні для виконання семестрового індивідуального завдання та проводяться контрольні заходи (тести) з поточної оцінки самостійної роботи студентів з засвоєння поточного матеріалу. Останні є складовою частиною поточного рейтингу і проводяться тільки у день проведення відповідних лекцій та лабораторних робіт. Система оцінювання орієнтована на отримання балів за своєчасність виконання студентами лабораторних та контрольних робіт, а також виконання завдань, які здатні розвинути практичні уміння та навички;
- правила поведінки на заняттях: студент повинен брати участь у розв'язку задач, готувати короткі доповіді;
- захист лабораторних робіт – захист відбувається у визначені терміни під час аудиторних занять;
- політика щодо академічної доброчесності– політика та принципи академічної доброчесності визначені у розділі 3 Кодексу честі Національного технічного університету України «Київський політехнічний інституті мені Ігоря Сікорського».

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

Поточний контроль проводиться методом опитування студентів під час приймання лабораторних робіт. Лабораторні роботи оцінюються виходячи з максимальної кількості балів за виконання кожного завдання:

- «відмінно» – 95 відсотків максимального балу;
- «добре» – 75-95;
- «задовільно» – 60-75;
- «достатньо» – 50 відсотків – робота виконана, але не захищена.

Календарний контроль: провадиться двічі на семестр як моніторинг поточного стану виконання вимог силабусу. Умовою позитивного першого календарного контролю є отримання не менше 25 балів, другого – отримання не менше 45 балів.

Модульна контрольна робота оцінюється із 40 балів. Проводиться у першому семестрі у вигляді тесту, у другому – написання практичних завдань. Приклади завдань наведено у Додатках А та Б відповідно.

Семестровий контроль: залік (1 та 2 семестри), також у 2-му семестрі передбачено захист курсової роботи.

Умови допуску до семестрового контролю: зарахування усіх лабораторних робіт, семестровий рейтинг більше 40 балів.

Рейтинг студента з кредитного модуля складається з балів, що він отримує за:

- виконання лабораторних робіт;
- виконання контрольних робіт.

За період вивчення дисципліни студент може набрати 100 балів. Їх розподіл між видами робіт наведено в таблиці 1 для першого семестру та таблиці 2 для другого семестру.

Таблиця 1 – Розподіл балів за Кредитний модуль 1

Номер лабораторної роботи								Разом	Контрольна робота	Всього
1	2	3	4	5	6	7	8			
7	8	7	7	8	8	8	7	60	40	100

Таблиця 2 – Розподіл балів за Кредитний модуль 2

Номер лабораторної роботи						Разом	Контрольна робота	Всього
9	10	11	12	13	14			
10	10	10	10	10	10	60	40	100

Студент, який у семестрі отримав менше 60 балів, може взяти участь у заліковій контрольній роботі. Контрольне завдання цієї роботи складається з трьох запитань з переліку, що наданий у цьому документі.

Кожне запитання оцінюється з 10 балів за такими критеріями:

- «відмінно» – повна відповідь (не менше 90% потрібної інформації), надані відповідні обґрунтування та особистий погляд – 10...9 балів;
- «добре» – достатньо повна відповідь (не менше 75% потрібної інформації), що виконана згідно з вимогами до рівня «умінь», або незначні неточності) – 8...7 балів;
- «задовільно» – неповна відповідь (не менше 60% потрібної інформації, що виконана згідно з вимогами до «стереотипного» рівня та деякі помилки) – 6...5 балів;
- «незадовільно» – незадовільна відповідь – 0 балів.

У цьому разі остаточний результат складається із балів, що отримані на заліковій контрольній роботі, та балів, отриманих протягом семестру.

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 питань, які виносяться на семестровий контроль (залікову контрольну роботу):
  1. Які базові типи даних реалізовані в мові програмування C?
  2. Область допустимих значень для `char` і `unsigned char`?
  3. Для чого використовується модифікатор `register`?
  4. Назвіть три форми завдання констант типу `char`.
  5. Дайте визначення масиву.
  6. Як ініціювати двомірний масив.
  7. За допомогою яких конструкцій здійснюється звернення до елементів структур?
  8. Чим відрізняється визначення структури від оголошення структури?
  9. Як ініціюються оголошення структур?
  10. Які існують особливості розподілу пам'яті для елементів об'єднань, які існують у цьому об'єднанні?
  11. Для чого слугує ключове слово `typedef`?
  12. Що таке вказівник?
  13. Якого типу можуть бути показники?
  14. За допомогою якої функції виконується форматове виведення даних на дисплей?
  15. Які формати використовуються для виводу цілих значень?
  16. Що відбувається, якщо довжина виведеного значення перевищує довжину, задану в форматі?
  17. Для чого в форматі задається знак мінус?
  18. Які формати використовуються для видачі значень з плаваючою точкою?
  19. Призначення функції `sprintf()`?
  20. Які існують особливості завдання аргументів функції `scanf()`?
  21. За допомогою якої функції здійснюється посимвольне введення даних з клавіатури?
  22. Перечисліть арифметичні операції в порядку спадання їх старшинства.
  23. Які математичні функції існують в бібліотеці мови C?
  24. Як працюють операції відношення?
  25. Які логічні операції реалізовані в мові C?
  26. Чим відрізняються порозрядні логічні операції від логічних операцій?
  27. Які існують різновиди операції присвоєння?
  28. Для чого можна використовувати операцію "умова"?
  29. Які допускаються операції з показниками?
  30. За допомогою якої операції здійснюється звернення до елемента структури через її адресу?
  31. За допомогою якої операції здійснюється вихід із перемикача? Що повинно слідувати за ключовим словом `case`?
  32. Які різновиди операторів циклу існують в мові C?
  33. Для чого використовується в циклах оператор `break`?
  34. Чи можна задавати в операторі `for` зміну двох параметрів?
  35. З яких складових частин складається програма на мові C?
  36. Чи допускаються в мові C вкладені цикли?
  37. Що таке прототип функції?
  38. Які властивості мають об'єкти з модифікатором `static`?
  39. Як відбувається розподіл пам'яті для зовнішніх визначень об'єкта?
  40. Механізм передачі параметрів функціям, реалізований в мові C?
  41. Як передавати масиви в функції?

42. Якими способами можна передавати дані між функціями?
44. Що надає можливість рекурсивного виклику функцій?
45. Які існують способи ініціалізації масивів рядків?
46. Якими засобами вводяться рядки символів?
47. Перечисліть функції обробки рядків.
43. Які існують особливості представлення масивів рядків з використанням масивів показників?
44. За рахунок чого може збільшуватися ефективність операцій обробки рядків при використанні масиву показників?
45. Дані. Константи. Рядкові константи. Спеціальні рядкові константи. Визначені іменовані константи.
46. Структура програми на мові C. Етапи перетворення вихідного модуля у модуль, який виконується.
47. Загальні поняття о структурованих типах даних. Масиви, структури, об'єднання.
48. Внутрішнє представлення даних. Данні типу int, char, float.
49. Представлення дійсних чисел в ЕОМ. Погрішність округлення . Поняття „ машинного нуля”.
50. Особливості операцій з вказівниками при роботі з масивами.
51. Алгоритм пошуку. Метод бінарного пошуку.
52. Оператори циклу. Організація затримок за допомогою покрокового циклу.
53. Сортування масивів.
54. Алгоритм злиття упорядкованих масивів.
55. Оператор перемикач.
56. Керуючі оператори розриву та подовження. Схеми їх виконання у циклах.
57. Алгоритм перетворення числа з одної системи числення у іншу.
58. Введення-виведення даних. Визначені 5 потоків у C.
59. Ітераційні алгоритми.
60. Типи форматів введення/виведення, модифікатори.
61. Читання символів з стандартного потоку stdin, вивід символів у потік stdout.
62. Символьні масиви, рядки. Символьні та строкові функції. Введення та виведення рядків.
63. Функції, структури визначення. Специфікація параметрів, тип функції.
64. Передача змінних у функціях, передача по значенню.
65. Передача змінних у функціях по адресу. Значення, що повертає функція.
66. Особливості передачі масивів та рядків у функціях.
67. Рекурсивні функції.
- Є можливість зарахування сертифікатів проходження дистанційних чи онлайн курсів Coursera за відповідною тематикою – зараховується додатково до 10 балів до загального рейтингу студента, якщо студент набрав не менше 75 балів за період вивчення курсу та отримав відповідний сертифікат.

### **Робочу програму навчальної дисципліни (силабус):**

**Складено** старшим викладачем Романовим Валерієм Володимировичем

**Ухвалено** кафедрою СП (протокол № 10 від 30.06.2022)

**Погоджено** Методичною комісією НН ІПСА (протокол № 8 від 17.06.2022)

**Приклади завдань модульної контрольної роботи (кредитний модуль 1)**

1. Що буде надруковано?

```
#include <stdio.h>
int main()
{
    int x=1, y=2;
    int *p1, *p2 = NULL;
    p1=&x;
    y=*p1;
    *p1+=10;
    if (p2==NULL) p2=p1;
    ++*p1;
    printf("x=%d y=%d *p2=%i\n",x,y,*p2);
    return 0;
}
```

2. Які значення будуть присвоєні змінним a,b,c,d,e,f?

```
#include<stdio.h>
int main()
{
    int a, b, c, d, e, f, n=12, m=-5, k=2;
    a=n/m;
    b=n%m;
    c=n | m;
    d=n&m;
    e=n<<k;
    f=n>>k;
    return 0;
}
```

3. Чому дорівнює сума при введенні 2-х варіантів значень: а) i=1; б) i=11?

```
#include<stdio.h>
int main()
{
    int sum=0, i ;
    scanf ("%d", &i);
    do
    {
        sum=sum+i;
        i++;
    }
    while (i<=10);
    printf ("sum=%d", sum);
    return 0;
}
```



4. Які з операцій в мові C недопустимі?

```
int m[10], *p;  
m=p;          // 1  
p=m;          // 2  
m++;          // 3  
p++;          // 4  
p=&m; // 5  
p=&m[0];      // 6
```

5. Що буде надруковано при введенні рядка "student"?

```
#include<stdio.h>  
int main()  
{  
    int n;  
    char *p1,*p2;  
    p1=new char[30];  
    scanf("%s", p1);  
    p2=p1;  
    while(*(++p1));  
    printf("%c %c\n  ", *p1,*p2);  
    n=p1- p2;  
    printf("n=%d",n);  
    delete[] p1;  
    return 0;  
}
```

6. Що буде надруковано?

```
#include<stdio.h>  
int main()  
{  
    int x=10;  
    printf("x=%d  ",x);  
    {  
        int x=12;  
        printf("x=%d  ",x);  
    }  
    printf("x=%d  ",x);  
    while (x++ <13)  
    {  
        int x=50;  
        x++;  
        printf("x=%d  ",x);  
    }  
    return 0;  
}
```

7. Що буде надруковано?

```
#include <stdio.h>
f(int a)
{
    auto int b=0;
    static c=3;
    b=b+1;
    c=c+1;
    return (a+b+c);
}
int main()
{
    int i, a=2;
    for(i=0; i<3; i++)
        printf("%d ", f(a));
}
```

8. Що буде надруковано?

```
#include <stdio.h>
int x=3;
int main(void)
{
    register int x=1;
    printf("\nx=%d",x);
    {
        extern int x;
        printf("\nx=%d",x);
    }
    printf("\nx=%d",x);
    return 0;
}
```

9. Які значення будуть присвоєні змінним a,b,d?

```
#include<stdio.h>
int main()
{
    int a=12, b, c=5, d;
    a+=a-=a*a;
    (b=3*5, b+4), b+5;
    d=2+(c+=c++,c+8,++c);
    printf("\na=%d, b=%d, d=%d",a,b,d);
    return 0;
}
```

**Приклади завдань модульної контрольної роботи (кредитний модуль 2)**

1. В квадратній матриці переставити рядки таким чином, щоб кількість нульових елементів в рядках зростала від першого рядка к останньому.
2. В заданому реченні знайти саме коротке і саме довге слово.
3. Нехай  $a$  та  $b$  – деякі цілі числа,  $a \leq b$ . Треба вивести всі цілі числа від  $b$  до  $a$  включно в порядку зменшення:  $b, b-1, \dots, a$ , використовуючи рекурсивний та ітераційний варіанти.
4. В прямокутній матриці переставити стовпці таким чином, щоб сума елементів стовпця зростала від першого стовпця к останньому.
5. Реалізувати функцію в програмі, в результаті звернення до якої з першого заданого рядка вилучається кожний символ, який належить і другому заданому рядку.
6. Обчислити суму всіх цілих чисел в діапазоні від  $a$  до  $b$ , використовуючи рекурсивний та ітераційний методи.
7. Задана цілочисельна матриця. Скласти програму, яка перетворює цю матрицю за правилом: елементи рядків матриці циклічно зсуваються на  $K$  позицій ліворуч. Циклічний зсув елементів матриці оформити у вигляді окремої функції.
8. Дано відрізки  $a, b, c, d$ . Для кожної трійки цих відрізків, з яких можливо побудувати трикутник, визначити величину площі даного трикутника. Визначити функцію  $PL()$ , яка друкує площину трикутника зі сторонами  $x, y, z$ , якщо такий трикутник існує.
9. Вивести символи введеного рядка у оберненому порядку, використовуючи ітераційний та рекурсивний варіанти.