

Київський національний університет
будівництва і архітектури
Кафедра інформаційних технологій

Шифр спеціальності	Назва спеціальності, освітньої програми	Освітній рівень
F7	Комп'ютерна інженерія, Комп'ютерні системи та мережі	Бакалавр

«Затверджую»

Завідувачка кафедри

Тетяна ГОНЧАРЕНКО / _____ /

Розробник силабуса

Олександр ПОПЛАВСЬКИЙ / _____ /



СИЛАБУС

Алгоритмізація та програмування

(назва освітньої компоненти (дисципліни))

1) Статус освітньої компоненти: обов'язкова	
2) Контактні дані викладача: (д.т.н., доцент Поплавський О.А., apoplavskyi@gmail.com , poplavskyi.oa@knuba.edu.ua , https://www.knuba.edu.ua/elementor-161621/ , +380 (93) 634 11 23)	
3) Пререквізити: Вища математика	
4) Коротка анотація дисципліни Навчальна дисципліна "Програмування і алгоритмічні мови" забезпечує студентів базовими знаннями та навичками в області алгоритмізації та програмування, необхідними для розробки ефективних комп'ютерних програм. Ця дисципліна спрямована на формування системного підходу до розв'язання задач, розвиток логічного мислення та освоєння основних принципів програмування. Особлива увага приділяється алгоритмічному мисленню та програмуванню з урахуванням особливостей комп'ютерних систем і середовищ виконання програм.	
5) Структура курсу: лекції, лабораторні роботи, самостійні роботи, РГР, екзамен	
Загальна кількість кредитів ECTS	4,5
Сума годин:	135
Вид індивідуального завдання	РГР
Форма контролю	Екзамен
6) Зміст курсу: Змістовий модуль 1. Основи програмування та алгоритмізації Лекція 1. Комп'ютер як основний інструмент програмування. Будова пам'яті комп'ютера. Програчне забезпечення. Лекція 2. Мови програмування. C, C ++. Лекція 3. Поняття алгоритму та правила його складання. Блок-схеми алгоритмів. Застосування циклів з лічильником Лекція 4. Структура програми на мові C, C++. Елементи мов програмування C, C++.	

Оператори та операнди.

Лекція 5. Структура мови C++. Змінні, оператори, модулі.

Змістовий модуль 2: Основи роботи з мовою C++

Лекція 6. Операції мов програмування C, C++. Арифметичні операції.

Лекція 7. Операції порівняння та логічні операції. Порозрядні (побітові) операції. Унарні операції.

Лекція 8. Функції вводу-виводу C, C++. Бібліотека `conio.h`. C++ введення виведення потоків.

Лекція 9. Циклічні процеси. Цикл `while`. Цикл `do`. Цикл `for`.

Лекція 10. Масиви. Застосування масивів. Сортування. Вказівники та масиви.

Лекція 11. Функції користувача. Підпрограми. Види підпрограм. Функції на мові C/C ++. Формальні і фактичні параметри. Масиви як параметри функцій. Класи пам'яті.

Змістовий модуль 3: Робота з графікою, текстами, структурами та файлами

Лекція 12. Підключення графічного режиму. Прості фігури. Малювання із застосуванням класів.

Лекція 13. Графіка VS C++. Малювання у консольному вікні. Функції малювання. Малювання геометричних фігур.

Лекція 14. Тип даних «текстовий рядок». Дії з рядками.

Лекція 15. Визначення та види рекурсії. Датчик псевдовипадкових чисел. Багатофайлові програми. Стандартні функції різного призначення.

Лекція 16. Структури. Використання структур. Оголошення і визначення структур. Доступ до членів структур. Ініціалізація структур. Масиви структур.

Лекція 17. Робота з файлами. Робота з текстовими файлами за допомогою файлових потоків. Створення потоків. Перевірка помилок виконання файлових операцій. Символьне введення / виведення.

Лекція 18. Створення проєктів. Початок роботи над проєктом. Компоненти. Події. Функція обробки події.

Лабораторні роботи

Змістовий модуль 1

Лабораторна робота №1. Робота з IDE Microsoft Visual Studio.

Лабораторна робота №2. Програма, що розгалужується

Лабораторна робота №3. Циклічні структури

Змістовий модуль 2.

Лабораторна робота №4. Одновимірні масиви

Лабораторна робота №5. Функції. Варіативні функції.

Лабораторна робота №6. Багатовимірні масиви

Змістовий модуль 3.

Лабораторна робота №7. Структуровані типи даних

Лабораторна робота №8. Показники

Лабораторна робота №9. Форматоване введення/виведення даних

Лабораторна робота №10. Робота з рядками типу `char*`.

Самостійна робота:

1. Ознайомлення з налаштуваннями IDE для роботи з проєктами. Вивчення стандартних налаштувань компіляції та налагодження.
2. Вивчення умовних операторів (`if-else`, `switch`). Побудова блок-схем для програм із розгалуженнями.
3. Опрацювання роботи циклів `for`, `while`, `do-while`. Приклади застосування вкладених циклів.
4. Сортування масивів різними алгоритмами (бульбашкове, вибір, вставки). Робота з пошуком мінімуму/максимуму в масивах.
5. Типи параметрів функцій (формальні, фактичні). Використання функцій із передачею масивів.
6. Реалізація програм для обробки матриць (додавання, множення). Використання вкладених циклів для роботи з багатовимірними масивами.
7. Різновиди структурованих типів даних у C++. Практичне використання структур і масивів структур.
8. Робота з динамічною пам'яттю та покажчиками. Використання покажчиків для роботи з масивами та функціями.
9. Використання функцій для роботи з рядками (додавання, порівняння, копіювання). Особливості роботи з рядками у C++ в порівнянні з бібліотекою `std::string`.
10. Основи роботи з текстовими файлами за допомогою файлових потоків. Методи перевірки помилок при роботі з файлами.
11. Основи роботи з графікою у VS C++. Реалізація простих графічних програм із малюванням геометричних фігур.
12. Підготовка базових компонентів і подій для роботи з проєктами. Реалізація функцій обробки подій.

Індивідуальна завдання: Виконання РГР згідно індивідуального варіанту для закріплення теоретичних знань і практичних навичок з ПтАМ у 1 та 2 семестрі.

Завдання до розрахунково-графічної роботи (РГР)

У межах розрахунково-графічної роботи здобувач вищої освіти повинен **самостійно виконати індивідуальне завдання з програмування**, що полягає у **розробці алгоритму розв'язання прикладної задачі та його програмній реалізації** з використанням однієї з алгоритмічних мов програмування.

У ході виконання РГР необхідно:

1. **Сформулювати прикладну задачу** відповідно до обраного варіанта.
2. **Розробити алгоритм розв'язання задачі**, описавши його у вигляді блок-схеми або псевдокоду.
3. **Проаналізувати алгоритм** з точки зору коректності та обчислювальної складності.
4. **Реалізувати алгоритм засобами алгоритмічної мови програмування** (C/C++, Java, Python або іншої, визначеної викладачем).

5. **Організувати роботу з вхідними та вихідними даними**, передбачивши перевірку коректності введення.
6. **Забезпечити модульну структуру програми** з використанням функцій (процедур).
7. **Провести тестування програми** на контрольних прикладах і проаналізувати отримані результати.
8. **Оформити програмний код відповідно до вимог структурного програмування** та стандартів оформлення.
9. **Зробити висновки** щодо ефективності обраного алгоритму та можливостей його вдосконалення.

Результатом виконання РГР є:

- пояснювальна записка з описом задачі, алгоритму та результатів тестування;
- програмний код реалізованого алгоритму;
- графічні матеріали (блок-схеми алгоритмів, таблиці тестових даних).

Варіанти розрахунково-графічної роботи (C++)

Варіант 1. Розробити програму на C++ для обробки одновимірного масиву: знайти мінімальний і максимальний елементи та їх позиції.

Варіант 2. Реалізувати програму на C++ для сортування масиву методом вибору та порівняти кількість операцій із методом обміну.

Варіант 3. Розробити програму для пошуку елемента в масиві з використанням лінійного та бінарного пошуку.

Варіант 4. Створити програму на C++ для транспонування квадратної матриці розміру $N \times N$.

Варіант 5. Реалізувати алгоритм обчислення суми елементів головної та побічної діагоналей матриці.

Варіант 6. Розробити програму для обробки двовимірного масиву: визначити рядок з максимальною сумою елементів.

Варіант 7. Створити програму на C++ для роботи з рядками: підрахунок кількості слів у введеному тексті.

Варіант 8. Реалізувати програму для перевірки симетричності рядка (паліндром).

Варіант 9. Розробити програму для роботи з файлами: запис масиву чисел у файл та зчитування з файлу з подальшою обробкою.

Варіант 10. Створити програму на C++ для пошуку найчастіше зустрічаємого елемента масиву.

Варіант 11. Реалізувати рекурсивний алгоритм обчислення факторіала та порівняти його з ітеративним.

Варіант 12. Розробити рекурсивну програму для обчислення чисел Фібоначчі.

Варіант 13. Створити програму для видалення елемента з масиву за заданим індексом.

Варіант 14. Реалізувати програму на C++ для злиття двох відсортованих масивів в один відсортований.

Варіант 15. Розробити програму для обчислення середнього арифметичного, дисперсії та стандартного відхилення масиву.

Варіант 16. Створити програму для обробки текстового файлу: підрахунок кількості рядків, слів і символів.

Варіант 17. Реалізувати програму для пошуку найбільшого спільного дільника (НСД) двох чисел алгоритмом Евкліда.

Варіант 18. Розробити програму на C++ для перевірки правильності розміщення дужок у виразі.

Варіант 19. Створити програму для роботи зі структурами: зберігання та обробка даних про студентів.

Варіант 20. Реалізувати програму для обчислення та аналізу часової складності алгоритму сортування (експериментально).

7) Посилання на сторінку електронного навчально-методичного комплексу дисципліни: <https://org2.knuba.edu.ua/course/view.php?id=254>