

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ЄРЕВАНСЬКИЙ НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ**



ЗАТВЕРДЖУЮ
Директор Єреванського навчально-наукового інституту ЗУНУ

Саак ГУДРАТЯН
2022 р.



ЗАТВЕРДЖУЮ

Перший проректор

Микола ПІНКАРИК

2022 р.

РОБОЧА ПРОГРАМА

з дисципліни

«Основи програмування»

ступінь вищої освіти – **бакалавр**

галузь знань - 12 Інформаційні технології

спеціальність - 123 Комп'ютерна інженерія

освітньо-професійна програма – «Комп'ютерна інженерія»

Кафедра фундаментальних дисциплін

Форма навчання	Курс	Семестр	Лекції (год.)	Лабораторні (год.)	ІРС (год.)	Тренінг (год.)	Самост. робота студ.(год.)	Разом (год.)	Іспит (сем.)
Заочна	1	1	8	4	-	-	168	180	2

Єреван- ЄННІ ЗУНУ – 2022

Робоча програма складена на основі освітньо-професійної програми «Комп'ютерна інженерія» підготовки здобувачів вищої освіти на першому (бакалаврському) рівні за спеціальністю 123 «Комп'ютерна інженерія» галузі знань 12 Інформаційні технології, затвердженої Вченою радою ЗУНУ (протокол № 9 від 15.06.2022р.).

Робочу програму розробила к.тех.н., доцент., доцент кафедри фундаментальних дисциплін ЄННІ ЗУНУ Анна ОВАКІМЯН.

Робоча програма затверджена на засіданні кафедри фундаментальних дисциплін, протокол № 1 від 26.08.2022 року.

Завідувач кафедри
фундаментальних дисциплін,
доктор технічних наук, професор



Гамлет АРУТЮНЯН

Розглянуто та схвалено групою забезпечення спеціальності 123 «Комп'ютерна інженерія», протокол № 1 від 30.08.2022 р.

Голова ГЗС,
д.т.н., професор



Гамлет АРУТЮНЯН

Гарант
д.т.н., професор



Гамлет АРУТЮНЯН

СТРУКТУРА РОБОЧОЇ ПРОГРАМИ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

"Основи програмування"

1. Опис дисципліни "Основи програмування"

Дисципліна - Основи програмування	Галузь знань, спеціальність, ступінь вищої освіти	Характеристика навчальної дисципліни
Кількість кредитів – 6	Галузь знань: 12 «Інформаційні технології»	Статус дисципліни: нормативна Мова навчання: українська, вірменська
	Спеціальність: 123 «Комп'ютерна інженерія»;	Рік підготовки: 1 Семестр: 1
Кількість змістовних модулів - 4	Ступінь вищої освіти: бакалавр	Лекції: 8 год.; Лабораторні роботи: 4 год.
Загальна кількість годин - 180		Самостійна робота: 168 год
		Вид підсумкового контролю – іспит

2. Мета й завдання вивчення дисципліни "Основи програмування"

2.1. Мета вивчення дисципліни

Програма та тематичний план дисципліни орієнтовані на глибоке та ґрунтовне засвоєння студентами основних алгоритмів розв'язку типових задач та їх реалізація за допомогою конкретних алгоритмічних мов програмування. Ця дисципліна відноситься до фундаментальних наук, які формують фаховий світогляд майбутніх програмістів. Курс "Основи програмування" охоплює методологічні основи алгоритмізації, методи та прийоми вивчення інформаційних систем, їх моделювання та реалізацію тощо. Названий курс повинен сприяти формуванню висококваліфікованих фахівців у галузі інформаційних технологій.

2.2. Завдання вивчення дисципліни

Головним завданням дисципліни є набуття студентами теоретичних знань про принципи побудови сучасних мов програмування, вивчення основних конструкцій мов, типових задач алгоритмічного програмування, набуття досвіду роботи в інтегрованому середовищі розробки програм (на прикладі компілятора Visual Studio C++) та проходженні повного циклу розробки (розробка алгоритму - кодування - компіляція - відлагодження - документування). Таке поєднання теоретичних та практичних знань та вмінь сприяє як достатньому оволодінню роботою з конкретними існуючими середовищами розробки, так і полегшенню освоєння нових засобів програмування та переходу на нові середовища та операційні системи в майбутньому.

Вивчення курсу "Основи програмування" передбачає наявність систематичних та ґрунтовних знань отримані при вивченні шкільного курсу "Основи інформатики.", цілеспрямованої роботи над вивченням спеціальної літератури, активної роботи на лекціях, практичних та лабораторних заняттях, самостійної роботи та виконання індивідуальних завдань.

2.3 Найменування та опис компетентностей, формування котрих забезпечує вивчення дисципліни

ЗК2. Здатність вчитися і оволодівати сучасними знаннями.

ЗК3. Здатність застосовувати знання у практичних ситуаціях.

СК2. Здатність використовувати сучасні методи і мови програмування для розроблення алгоритмічного та програмного забезпечення.

2.4 Передумови для вивчення дисципліни

Зазначена дисципліна включена до циклу дисциплін професійної підготовки за переліком програми. У структурно-логічній схемі навчання зазначена дисципліна розміщена на І-му курсі. Вивчення курсу передбачає наявність систематичних та ґрунтовних знань із курсів «Дискретна математика », цілеспрямованої роботи над вивченням спеціальної літератури, активної роботи на лекціях та лабораторних заняттях, самостійної роботи.

2.4. Результати навчання

ПРН3. Знати новітні технології в галузі комп'ютерної інженерії.

ПРН6. Вміти застосовувати знання для ідентифікації, формулювання і розв'язування технічних задач спеціальності, використовуючи методи, що є найбільш придатними для досягнення поставлених цілей.

ПРН8. Вміти системно мислити та застосовувати творчі здібності до формування нових ідей.

ПРН10. Вміти розробляти програмне забезпечення для вбудованих і розподілених застосувань, мобільних і гібридних систем, розраховувати, експлуатувати, типове для спеціальності обладнання.

ПРН12. Вміти ефективно працювати як індивідуально, так і у складі команди.

ПРН16. Вміти оцінювати отримані результати та аргументовано захищати прийняті рішення.

ПРН18. Використовувати інформаційні технології та для ефективного спілкування на професійному та соціальному рівнях.

ПРН20. Усвідомлювати необхідність навчання впродовж усього життя з метою поглиблення набутих та здобуття нових фахових знань, удосконалення креативного мислення.

3. Програма навчальної дисципліни "Основи програмування"

Змістовний модуль 1. Основні поняття програмування.

Тема 1. Основні поняття та означення

Поняття архітектури комп'ютера. Архітектури комп'ютера фон Неймана. Архітектура системи команд. Інформація в пам'яті комп'ютера. Типи комп'ютерів. Програмне забезпечення. Засоби створення програм. Поняття алгоритму. Властивості алгоритмів, форми їх представлення. Основні структури алгоритмів.

Тема 2. Синтаксис мови C++

Множина символів C++ та множина представимих символів. Правила формування констант, ідентифікаторів. Ключові слова. Використання коментарів у програмах. Поняття лексеми.

Змістовний модуль 2. Основні структури в програмуванні.

Тема 3. Структура програми на мові C++

Склад програми на мові C++. Вихідні файли програми. Виконання програм. Особливості функції `main()`. Поняття "час життя" і "область видимості". Приклади найпростіших програм.

Тема 4. Оголошення змінних

Базові типи даних. Правила приведення базових типів даних. Модифікатори змінних. Автоматичні змінні. Реєстрові змінні. Зовнішні змінні та функції статичні змінні. Змінні класу `volatile`. Новий стиль заголовків.

Тема 5. Керування вводом-виводом на екран

Основні оператори вводу та виводу на мові C++, зокрема на прикладі `printf()` та `scan()`. Формат вводу-виводу різних типів даних (чисел, рядків, вказівників тощо).

Тема 6. Вирази

Арифметичні операції. Оператор присвоювання `.`. Поняття виразу. Оператори інкремента і декремента. Оператор `sizeof`. Порозрядні логічні операції. Операції зсуву вліво і вправо. Оператори порівняння. Операція "кома". Пріоритет і порядок виконання операцій.

Змістовний модуль 3. Засоби реалізації основних структур в програмуванні .

Тема 7. Засоби мови C++ для реалізації основних структур алгоритмів
Умовні оператори. Оператори if. Оператори if-else. Умовний оператор ?:.
Оператор switch. Оператори циклу. Цикли for. Цикли while. Цикли do-while.
Оператор break. Оператор continue. Оператор goto і метки.

Тема 8. Складні типи даних в C++
Масиви. Ініціалізація масивів. Багатомірні масиви. Масиви як параметри функцій. Структури й операції з ними. Структури як аргументи функцій. Масиви структур. Показчики на структури. Передача по посиланню членів масивів структур. Об'єднання й операції з ними.

Тема 9. Вказівники та посилання
Загальний огляд. Розмінування вказівників. Арифметика вказівників.
Вказівники. на вказівники. Вказівники. на функції. Посилання. Передача параметрів за посиланням і за значенням. Використання вказівників і посилань із ключовим словом const.

Змістовний модуль 4. Функції та робота з строковими змінними.

Тема 10. функції
Параметри й аргументи функцій. Аргументи за замовчуванням. Простір імен. Функції, що вбудовуються (inline-). Рекурсивні функції. Математичні функції. Функції округлення.

Тема 11. Рядки і операції з ними
Масиви символів у C++. Визначення довжини рядків. Копіювання і конкатенація рядків. Порівняння рядків. Перетворення рядків. Звертання рядків. Пошук символів.

Тема 12. Використання стандартного класу стрічок string
Конструктори рядків. Зміна величини рядка і її розміру. Присвоювання, додавання і обмін рядків. Доступ до символів рядка. Копіювання рядків і порядків. Порівняння рядків. Операції пошуку. Вставка символів у рядок. Заміна і видалення символів з рядка. Операції вводу-виводу рядків.

Тема 13. Файловий ввід / вивід
Взаємодія з файлами. Типи файлів. Функції стандартного вводу / виводу. Функції довільного доступу до файлу.

СТРУКТУРА ЗАЛІКОВОГО КРЕДИТУ ДИСЦИПЛІНИ "ОСНОВИ ПРОГРАМУВАННЯ"

заочна форма навчання	Кількість годин		
	Лекції	Лабора торні заняття	Самостійн а робота
Змістовний модуль 1. Основні поняття програмування			
Тема 1. Основні поняття та означення	2	1	6
Технологічний цикл розробки програм			6
Тема 2. Синтаксис мови C++			6
Написання програм в середовищі MS Visual Studio.			6
Змістовний модуль 2. Основні структури в програмуванні			
Тема 3. Структура програми на мові C++	2	1	6
Тема 4. Оголошення змінних			6
Програмування простих задач			6
Тема 5. Керування вводом-виводом на екран			6
Адресна арифметика			6
Тема 6. Вирази			8
Арифметичні вирази у мові C/C++			6
Програмування циклічних процесів			6
Змістовний модуль 3. Засоби реалізації основних структур в програмуванні			
Тема 7. Засоби мови C++ для реалізації основних структур алгоритмів	2	1	6
Логічні вирази у мові C/C++			6
Побітові вирази у мові C/C++			8
Тема 8. Складні типи даних в C++			6
Алгоритми з циклічною структурою			6
Робота з масивами			6
Тема 9. Вказівники та посилання			6
Робота з даними типу "структура"			6
Змістовний модуль 4. Функції та робота з строковими змінними			
Тема 10. функції	2	1	6
Використання функцій.			8
Тема 11. Рядки і операції з ними			6
Тема 12. Використання стандартного класу стрічок string			6
Обробка стрічкових даних			6
Тема 13. Файловий ввід / вивід			6
Ввід-вивід у файл.			6
Разом	8	4	168

5. ТЕМАТИКА ЛАБОРАТОРНИХ ЗАНЯТЬ

Лабораторне заняття №1(4/1год.).

Тема: Написання програм в середовищі VS C++

Мета: Навчитись працювати в середовищі програмування VS C++.

Лабораторне заняття №2(5/1год.).

Програмування простих задач

Мета: Навчитися складати алгоритми та програмувати задачі невисокої складності

Лабораторне заняття №3(6/1год.).

Програмування циклічних процесів

Мета: Навчитися складати алгоритми та програмувати задачі з циклічними процесами

Лабораторне заняття №4(6/2год.).

Робота з масивами

Мета: Навчитися складати алгоритми та програмувати задачі, що використовують масиви.

Лабораторне заняття №5(6/2год.).

Робота з даними типу "структура"

Мета: Навчитися складати алгоритми та програмувати задачі елементами даних «структура».

Лабораторне заняття №6(6/1год.).

Використання функцій

Мета: Навчитися складати алгоритми та програмувати задачі з використанням функцій користувача.

Лабораторне заняття №7(6/1год.).

Обробка стрічкових даних

Мета: Навчитися складати алгоритми та програмувати задачі з даними представленими у вигляді стрічок.

Лабораторне заняття №8(6/1год.).

Ввід-вивід у файл

Мета: Навчитися складати алгоритми та програмувати задачі з файловими потоками

6. Самостійна робота

Для успішного вивчення і засвоєння дисципліни «Основи програмування» студенти повинні володіти значним обсягом інформації, частину якої вони отримують і опрацьовують шляхом самостійної роботи. Самостійна робота полягає в опрацюванні сучасної навчальної і наукової фахової літератури.

Результати самостійної роботи студента з дисципліни «Основи програмування» оцінюються за 100-бальною шкалою у вигляді аналітичного огляду наукових джерел або літератури відповідно до обраної з переліку теми. Алгоритм Евкліда . Реалізувати програму для знаходження найбільшого спільного дільника двох чисел за допомогою алгоритму Евкліда.

1. Сортування вставками . Написати програму для сортування масиву методом

вставок.

2. Метод підрахунку . Реалізувати програму для сортування масиву методом підрахунку.
3. Нерекурсивне обчислення числа Фібоначчі . Написати програму, що обчислює числа Фібоначчі без рекурсії.
4. Пошук найбільшого спільного кратного (НСД) . Реалізувати програму для знаходження НСК двох чисел.
5. Числа Армстронга . Написати програму для перевірки, чи є задане число числом Армстронга.
6. Числа щасливі та нещасливі . Реалізувати програму для перевірки, чи є число "щасливим" (наприклад, 7).
7. Сортування Шелла . Написати програму для сортування масиву за допомогою алгоритму сортування Шелла.
8. Реалізація алгоритму Лемера . Створити програму для генерації псевдовипадкових чисел за допомогою алгоритму Лемера.
9. Алгоритм сортування методом злиття . Реалізувати програму для сортування масиву методом злиття.
10. Генерація простих чисел (решето Ератосфена) . Написати програму для знаходження всіх простих чисел до заданого числа.
11. Ітеративна та рекурсивна реалізація обчислення факторіалу . Написати дві версії програми для обчислення факторіалу.
12. Аналіз алгоритму пошуку мінімального елемента в масиві . Реалізувати та порівняти різні методи пошуку мінімуму.
13. Реалізація кільцевого буфера . Написати програму для реалізації кільцевого буфера в масиві.

Ці теми більше орієнтовані на алгоритми, структури даних та нестандартні задачі, що можуть бути корисними для поглиблення знань з C++.допомогою вказівників.

14. Динамічна пам'ять . Створення програми для динамічного виділення пам'яті під масив.
15. Класи та об'єкти . Створення класу для опису студента (ім'я, прізвище, оцінки).
16. Конструктори та деструктори . Написати клас для управління масивом з конструктором і деструктором.
17. Наслідування . Розробити базовий клас "Транспортний засіб" та похідні класи "Автомобіль", "Мотоцикл".
18. Поліморфізм . Реалізувати поліморфізм для обчислення площі різних фігур (коло, квадрат).
19. Віртуальні функції . Написати програму з використанням віртуальних функцій для наслідуваних класів.
20. Абстрактні класи . Створити абстрактний клас "Тварина" та реалізувати похідні класи "Кіт", "Собака".

21. Шаблони функцій . Написати шаблон функції для обчислення мінімуму та максимуму двох елементів.
22. Шаблони класів . Реалізувати шаблон класу для збереження різних типів даних.
23. Обробка файлів . Створити програму для читання даних з файлу та їх обробки.
24. Запис у файл . Написати програму для запису даних у файл.
25. Ведення журналу (логування) у файл . Реалізувати програму для ведення логів у текстовому файлі.
26. Структури . Створити структуру для опису автомобіля (марка, рік випуску, ціна) та реалізувати пошук автомобіля з мінімальною ціною.
27. Перерахування (enum) . Використання перерахувань для створення програми вибору типу меню.
28. Об'єктно-орієнтоване програмування (ООП): інкапсуляція . Створити клас "Банк", що інкапсулює інформацію про баланс рахунку.
29. ООП: композиція . Реалізувати клас "Комп'ютер", що складається з класів "Процесор", "Пам'ять", "Диск".
30. ООП: агрегація . Створити клас "Факультет", що містить студентів як об'єкти.
31. ООП: наслідування із захищеними членами . Реалізувати клас "Співробітник" і похідні "Менеджер", "Програміст".
32. Робота з бібліотеками STL: вектори . Написати програму для сортування елементів вектора.
33. STL: списки . Реалізувати програму для додавання та видалення елементів зі списку.
34. STL: черги . Створити програму для симуляції черги клієнтів у банку.
35. STL: стеки . Реалізувати стек для перевірки коректності вкладених дужок у математичному виразі.
36. STL: множини та асоціативні контейнери . Використати множини для видалення дубльованих елементів із масиву.
37. STL: мапи . Створити мапу для збереження даних про студентів (ID - ім'я) та пошуку за ID.
38. Алгоритми STL . Написати програму для пошуку мінімального елемента в контейнері з використанням STL.
39. Робота з потоками введення-виведення . Реалізувати програму для роботи з двома потоками вводу та виводу.
40. Прототипування та тестування коду . Створення програми для тестування простих математичних функцій.
41. Обробка винятків . Реалізувати програму, яка перевіряє ділення на нуль та обробляє помилку.
42. Множинне наслідування . Написати клас "Людина", який наслідується від класів "Студент" та "Співробітник".
43. Алгоритм сортування бульбашкою . Реалізувати алгоритм сортування масиву за зростанням.
44. Алгоритм сортування вибором . Написати програму для сортування масиву за допомогою методу вибору.
45. Алгоритм сортування швидким методом (Quicksort) . Реалізувати алгоритм

швидкого сортування.

46. Бінарний пошук . Створити програму для пошуку елемента в відсортованому масиві за допомогою бінарного пошуку.

47. Робота з рекурсією: Ханойська вежа . Реалізувати алгоритм рішення Ханойської вежі.

48. Графи: пошук в глибину . Написати програму для реалізації алгоритму пошуку в глибину.

49. Графи: пошук в ширину . Реалізувати програму для обходу графа методом пошуку в ширину.

7. Засоби оцінювання та методи демонстрування результатів навчання.

У процесі вивчення дисципліни «Основи програмування» використовуються наступні засоби оцінювання та методи демонстрування результатів навчання:

- стандартизовані тести;
- поточне опитування;
- презентації результатів виконаних завдань та досліджень;
- студентські презентації та виступи на наукових заходах;
- інші види індивідуальних та групових завдань.

8. Критерії, форми поточного та підсумкового контролю

В процесі вивчення дисципліни «Основи програмування» використовуються наступні методи оцінювання навчальної роботи студента:

- стандартизовані тести;
- поточне опитування;
- презентації результатів виконаних завдань та досліджень;
- студентські презентації та виступи на наукових заходах;
- інші види індивідуальних та групових завдань.

Оцінювання здійснюється за 100 бальною шкалою.

Критерії оцінювання знань:

90–100 балів – у повному обсязі володіє навчальним матеріалом, вільно самотійно та аргументовано його викладає під час відповідей, глибоко та всебічно розкриває зміст питань;

75–89 балів – достатньо повно володіє навчальним матеріалом, але при викладанні деяких питань не вистачає достатньої глибини та аргументації, допускаються при цьому окремі несуттєві неточності та незначні помилки;

65–74 бали – в цілому володіє навчальним матеріалом та викладає його основний зміст, але без глибокого всебічного аналізу, обґрунтування та аргументації, допускаючи при цьому окремі суттєві неточності та помилки;

60–64 бали – не в повному обсязі володіє навчальним матеріалом, фрагментарно (без

аргументації та обґрунтування) його викладає, недостатньо розкриває зміст теоретичних питань, допускаючи при цьому суттєві неточності;

1–59 балів – не володіє навчальним матеріалом, не розкриває зміст теоретичних питань.

Шкала оцінювання:

За шкалою Університету	За національною шкалою	За шкалою ECTS
90-100	відмінно	A (відмінно)
85-89	добре	B (дуже добре)
75-84		C (добре)
65-74	задовільно	D (задовільно)
60-64		E (достатньо)
35-59	незадовільно	FX (незадовільно з можливістю повторного складання)
1-34		F (незадовільно з обов'язковим повторним курсом)

9. Інструменти, обладнання та програмне забезпечення, використання яких передбачає навчальна дисципліна

№	Найменування	Номер теми
1	Мультимедійний проектор	1-13
2	Проекційний екран	1-13
3	Комунікаційне програмне забезпечення (Internet Explorer, Google Chrome, Firefox)	1-13
4	Операційна система Windows, наявність доступу до мережі Internet	1-13
5	Персональні комп'ютери	1-13
6	Комунікаційне програмне забезпечення (Zoom) для проведення занять у режимі онлайн (за необхідності)	1-13
7	Комунікаційна навчальна платформа (Moodle) для організації дистанційного навчання (за необхідності)	1-13
8	Базове програмне забезпечення Microsoft Office	1-13
9	Спеціалізоване програмне забезпечення для моделювання та програмування: Microsoft Visio, MS Visual Studio. https://cpp.sh/ - онлайн-додаток для компіляції програм на мові програмування C++.	1-13

РЕКОМЕНДОВАНІ ДЖЕРЕЛА ІНФОРМАЦІЇ

1. Васильєв О. Характеристики Програмування C++ в прикладах і задачах. Навч. пос. Збільшений формат В5 Видавництво Ліра-К., 2020 382 с.

2. Пекарський Б.Г. Основи програмування: Навчальний посібник.- Кондор, 2018.-364 с.

3. C++ Crash Course: A Fast-Paced Introduction./ Lospinoso Josh. ISBN 1593278885. - 2019.-792с. 5. International Standard ISO/IEC 14882:2014(E) – Programming Language C++ , ISBN-13: 978-0321563842: [Електронний ресурс]. – Режим доступу: <https://isocpp.org/std/the-standard>.

4. C/C++ language and standard libraries reference: [Електронний ресурс]. – Режим доступу: <https://msdn.microsoft.com/en-us/library/hh875057.aspx>.

5. C++ Tutorial: [Електронний ресурс]. – Режим доступу <https://www.w3schools.com/cpp/>.
6. C++ Language Tutorials: [Електронний ресурс]. – Режим доступу <http://www.cplusplus.com/doc/tutorial/>
7. Learn C++ programming language : [Електронний ресурс]. – Режим доступу <https://www.tutorialspoint.com/cplusplus/index.htm>.
8. Уроки програмування на C++ : [Електронний ресурс]. – Режим доступу <https://acode.com.ua/uroki-po-cpp/>.
9. Основи програмування на C ++ для початківців : [Електронний ресурс]. – Режим доступу <https://purecodecpp.com/uk/>.
10. Навчальні матеріали для вивчення основ консольного програмування мовою C++: [Електронний ресурс]. – Режим доступу https://sites.google.com/site/zsuelearning/c_plus.