


МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

ЗАТВЕРДЖУЮ

В.о. декана факультету комп'ютерних
інформаційних технологій


_____ Ігор ЯКИМЕНКО
«__» _____ 20__ р.



ЗАТВЕРДЖУЮ

В.о. проректора
з науково-педагогічної роботи


_____ Віктор ОСТРОВЕРХОВ
«__» _____ 20__ р.

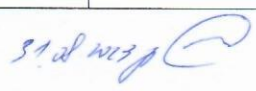


РОБОЧА ПРОГРАМА
з дисципліни
«ПРОГРАМУВАННЯ П»

Ступінь вищої освіти – бакалавр
Галузь знань – **12 Інформаційні технології**
Спеціальність – **125 Кібербезпека**
Освітньо-професійна програма – **Кібербезпека**

Кафедра кібербезпеки

Форма навчання	Курс	Семестр	Лекції (год.)	Лабор. (год.)	ІРС (год.)	Тренінг КПІЗ (год)	СРС (год.)	Разом (год.)	Екзамен (сем)
Денна	2	3	42	56	6	14	122	240	3



Робоча програма складена на основі освітньо-професійної програми підготовки бакалавра галузі знань 12 Інформаційні технології спеціальності – 125 Кібербезпека, затвердженої на засіданні Вченою радою ЗУНУ протокол № 9 від 15.06.2022р.

Робочу програму склав доцент кафедри кібербезпеки, к.т.н., доцент Івасьєв Степан Володимирович.

Робоча програма затверджена на засіданні кафедри кібербезпеки, протокол № 1 від 28.08.2023 р.

Завідувач кафедри
кібербезпеки



Василь ЯЦКІВ

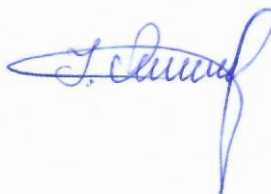
Розглянуто та схвалено групою забезпечення спеціальності кібербезпека та захист інформації, протокол № 1 від 30.08.2023 р.

Голова групи
забезпечення спеціальності



Василь ЯЦКІВ

Гарант ОП



Ігор ЯКИМЕНКО

СТРУКТУРА РОБОЧОЇ ПРОГРАМИ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

1. Опис дисципліни „Програмування П”

Дисципліна – «Програмування П»	Галузь знань, спеціальність, СВО	Характеристика навчальної дисципліни
Кількість кредитів – 7	Галузь знань 12 - Інформаційні технології	Статус дисципліни – обов’язкова Мова навчання - українська
Кількість залікових модулів – 4	Спеціальність - 125 Кібербезпека	Рік підготовки – 2 Семестр – 3
Кількість змістових модулів – 3	Ступінь вищої освіти – бакалавр	Лекції – 42 год. Лабораторні заняття – 56 год.
Загальна кількість годин – 240		СРС – 136 год, в т.ч. тренінг – 14год. Індивідуальна робота -6 год.
Тижневих годин: 17 год., з них аудиторних – 7 год.		Вид підсумкового контролю –екзамен

2. Мета й завдання вивчення дисципліни «Програмування П»

2.1. Мета вивчення дисципліни

Метою дисципліни є вивчення науково-практичного інструментарію технологій структурного та об’єктно-орієнтованого підходів при проектуванні комп’ютерних програм та його використанні при реалізації програмних додатків. Програма та тематичний план дисципліни орієнтовані на глибоке та ґрунтовне засвоєння студентами систематичних знань та практичних навичок для створення програмних продуктів з ефективним використанням сучасних технологій.

2.2 Завдання вивчення дисципліни

Завданням вивчення дисципліни є вивчення науково-практичного інструментарію проектування прикладних програмних додатків та їх складових за допомогою сучасних засобів автоматизованого проектування. В результаті вивчення дисципліни студенти мають вміти визначати алгоритми вирішення поставлених задач, проектувати загальну структуру програмних додатків, реалізовувати поставлені задачі за допомогою основних бібліотек та команд мови C++. Завдання проведення лекцій полягає у викладанні основних теоретичних положень сучасних об’єктно орієнтованих мов програмування та найкращих практик побудови алгоритмічних конструкцій. Завдання проведення практичних занять, як одна з основних форм навчального процесу, передбачає поглиблення розуміння і застосування на практиці основних методів побудови алгоритмічних конструкцій та реалізації найпоширеніших патернів об’єктно орієнтованого програмування.

2.3. Найменування та опис компетентностей, формування котрих забезпечує вивчення дисципліни

Здатність прогнозувати, виявляти та оцінювати стан інформаційної безпеки об’єктів і систем; знання основ структурного програмування; вміння визначати для вирішення задачі технологію програмування, мову, систему програмування, інструментальне середовище; вміння здійснювати функціональну та об’єктну декомпозицію програми відповідно до обраної технології програмування; навички розробки коду програми відповідно до структурної ідеології проектування програмних додатків; розуміння переваг та недоліків використання стандартних типів даних; знання циклічних операторів, операторів розгалуження та вміння їх ефективного застосування; володіння навичками обробки стрічкових даних; використання функцій для оптимізації коду програми; вміння створення функцій та класів для багаторазового використання при вирішенні типових задач; володіти методами та технологіями об’єктно-орієнтованого програмування; навички виявлення та виправлення помилок на ранньому етапі створення програмного коду.

2.4 Передумови для вивчення дисципліни.

Вивчення курсу „Програмування П” передбачає наявність систематичних та ґрунтовних знань із суміжних курсів («Дискретна математика», «Програмування I, а також цілеспрямованої роботи на лекційних та лабораторних заняттях, самостійної роботи студентів.

2.5. Результати навчання

Набуття навичок здійснювати оцінку можливості проникнення в ІТ системи та мережі шляхом експлуатації наявних вразливостей; оцінювати можливості та ефективність застосування, в тих чи інших умовах, інструментальних засобів оцінки вразливостей ІТ систем та мереж; володіти методам та засобами проектування програмних додатків; здійснювати декомпозицію програмної задачі; визначати алгоритм для вирішення окремих підзадач проектів; виділяти логічні та

синтаксичні помилки програмного коду; реалізувати поставлену задачу за допомогою основних бібліотек та команд мови C++.

3. Програма навчальної дисципліни «Програмування П»

Змістовий модуль 1. Основи об'єктно орієнтованого програмування.

Тема 1. Основи ООП в C++.

1. Класи, Об'єкти і Методи. 2. Специфікатори доступу public і private. 3. Інкапсуляція, Геттери і Сеттери.

Література: 1, 2, 3.

Тема 2. Конструктори.

1. Список ініціалізації членів класу. 2. Ініціалізація нестатичних членів класу. 3. Делегуючі конструктори. 4. Деструктори. 5. Прихований вказівник *this.

Література: 1, 3.

Тема 3. Можливості використання класів.

1. Класи і заголовкові файли. 2. Класи і const. 3. Статичні змінні-члени класу. 4. Статичні методи класу.

Література: 1, 2, 3.

Тема 4. Складні структури коду з використанням класів.

1. Анонімні об'єкти. 2. Вкладені типи даних в класах. 3. Вимірювання часу виконання (таймінг) коду.

Література: 1, 2, 3.

Тема 5. Перевантаження операторів в C++.

1. Перевантаження операторів. 2. Перевантаження операторів через дружні функції. 3. Перевантаження операторів через звичайні функції. 4. Перевантаження операторів вводу і виводу. 5. Перевантаження операторів через методи класу. 6. Перевантаження спеціальних операторів.

Література: 1, 2, 3.

Тема 6. Конструктор копіювання

1. Конструктор копіювання. 2. Копіююча ініціалізація. 3. Конструктори конвертації, ключові слова explicit і delete. 4. Перевантаження оператора присвоювання. 5. Поверхнєве і глибоке копіювання.

Література: 1, 5, 8.

Змістовий модуль 2. Ієрархія класів.

Тема 7. Зв'язки між об'єктами в C++.

1. Типи зв'язків між об'єктами. 2. Композиція об'єктів. 3. Агрегація. 4. Асоціація. 5. Залежність. 6. Контейнерні класи. 7. Список ініціалізації std::initializer_list.

Література: 2, 3,10.

Тема 8. Спадкування в C++.

1. Введення в Спадкування. 2. Базове спадкування. 3. Порядок побудови дочірніх класів. 4. Конструктори і ініціалізація дочірніх класів.

Література: 1, 2, 3.

Тема 9. Спадкування і специфікатори доступу.

1. Спадкування і специфікатор доступу protected. 2. Додання нового функціоналу в дочірній клас. 3. Перевизначення методів батьківського класу. 4. Приховування методів батьківського класу. 5. Множинне спадкування.

Література: 3, 6,13.

Тема 10. Віртуальні функції в C++.

1. Віртуальні функції в C++. 2. Вказівники, Посилання і Спадкування. 3. Віртуальні функції і Поліморфізм. 4. Модифікатори override і final. 5. Віртуальні деструктори і Віртуальне присвоювання. 6. Раннє і Пізнє зв'язування. 7. Віртуальні таблиці.

Література: 1, 2, 3.

Тема 11. Чисті віртуальні функції.

1. Чисті віртуальні функції. 2. Інтерфейси та Абстрактні класи. 3. Віртуальний базовий клас. 4. Обрізка об'єктів.

Література: 4, 5, 7,12.

Тема 12. Динамічне приведення типів.

1. Оператор dynamic_cast. 2. Вивід об'єктів класів через оператор виводу. 3. 4.

Література: 1, 2,11.

Тема 13. Шаблони в C++.

1. Екземпляри шаблонів функцій. 2. Шаблони класів. 3. Параметр non-type в шаблоні. 4. Явна спеціалізація шаблону функції та класу. 5. Часткова спеціалізація шаблонів і Вказівники

Література: 4, 7, 8.

Змістовий модуль 3. Спеціальний розділ.

Тема 14. Винятки в C++.

1. Обробка винятків. Оператори throw, try і catch. 2. Винятки, Функції і Розкручування стеку. 3. Неспіймані винятки і обробники catch-all. 4. Класи-винятки і Спадкування. 5. Повторна генерація винятків. 6. Функціональний try-блок.

Література: 7, 9.

Тема 15. Розумні вказівники і Семантика переміщення в C++

1. Розумні вказівники і Семантика переміщення в C++. 2. Посилання r-value. 3. Конструктор переміщення і Оператор присвоювання переміщенням. 4. Функція std::move(). 5. Розумний вказівник.

Література: 1, 3.

Тема 16. Стандартна бібліотека шаблонів (STL) в C++.

1. Контейнери STL. 2. Ітератори STL. 3. Алгоритми STL.

Література: 4, 7.

Тема 17. std::string в C++.

1. Рядкові класи std::string і std::wstring. 2. Створення, знищення і конвертація std::string. 3. Довжина і ємність std::string. 4. Доступ до символів std::string. Конвертація std::string в рядки C-style.

Література: 7, 8.

Тема 18. Потоки вводу і виводу.

1. Функціонал класу istream. 2. Функціонал класів ostream і ios. Форматування виводу. 3. Потокові класи і Рядки. 4. Стани потоку та валідація користувацького вводу. 5. Базовий файловий ввід і вивід. 6. Текстові файли. 7. Двійкові файли.

Література: 4, 6, 9.

4. Структура залікового кредиту дисципліни

	Кількість годин					Контрольні заходи
	Лекції і	Лабор. заняття	СРС	ІР С	Тренінг, КПЗ	
<i>Змістовий модуль 1 Основи об'єктно орієнтованого програмування</i>						
1. Основи ООП в C++.	2	3		6	1	Поточне опитування
2. Конструктори.	2	3		6		Поточне опитування
3. Можливості використання класів.	2	3		6		Поточне опитування
4. Складні структури коду з використанням класів.	4	3	1	6	1	Поточне опитування
5. Перевантаження операторів в C++.	2	3		7	1	Поточне опитування
6. Конструктор копіювання.	2	4		7	1	Поточне опитування
<i>Змістовий модуль 2. Ієрархія класів</i>						
7. Зв'язки між об'єктами в C++.	2	3		7	1	Поточне опитування
8. Спадкування в C++.	2	3		7		Поточне опитування
9. Спадкування і специфікатори доступу.	2	3	1	7	1	Поточне опитування
10. Віртуальні функції в C++.	2	3	1	7	1	Поточне опитування
11. Чисті віртуальні функції.	4	4		7	1	Поточне опитування
12. Динамічне приведення типів.	2	3	1	7	1	Поточне опитування
13. Шаблони в C++.	2	3		7	1	Поточне опитування
<i>Змістовий модуль 3. Спеціальний розділ</i>						

14. Винятки в C++.	4	3		7	1	Поточне опитування
15. Розумні вказівники. Семантика переміщення в C++.	2	3	1	7	1	Модуль
16. Стандартна бібліотека шаблонів (STL) в C++.	2	3		7	1	Поточне опитування
17. Std::string в C++.	2	3		7	1	Поточне опитування
18. Потоки вводу і виводу.	2	3	1	7		Поточне опитування
Разом	42	56	6	136	14	

5. Тематика лабораторних занять.

Лабораторна робота №1.

Тема: Вступ в об'єктно-орієнтоване програмування.

Мета: Ознайомитися з поняттям класу, навчитися проектувати класи та використовувати їх в вирішенні прикладних задач.

Питання для обговорення: Поняття класу та об'єкту Члени та методи класу

Література: 1, 2.

Лабораторна робота №2.

Тема: Класи. Конструктори та деструктори.

Мета: Вивчення основних методів проектування конструкторів та деструкторів.

Питання для обговорення: Можливості середовища. Засоби середовища розробки Visual Studio.

Література: 2, 4.

Лабораторна робота №3.

Тема: Дружні функції та класи.

Мета: Зрозуміти призначення дружніх функцій і класів.

Питання для обговорення: Механізм доступу до закритих даних класу. Відмінність функцій – друзів і методів класу.

Література: 1, 3, 9.

Лабораторна робота №4.

Тема: Перевантаження операцій та функцій.

Мета: Отримати практичні навички реалізації поліморфізму на прикладі перевантаження.

Питання для обговорення: Перевизначення операцій. Перевантаження функцій та приведення типів.

Література: 5, 6, 10.

Лабораторна робота №5.

Тема: Успадкування класів.

Мета: Отримати практичні навички по реалізації успадкування.

Питання для обговорення: Успадкування, Множинне спадкування. Використання віртуальних функцій.

Література: 3, 8.

Лабораторна робота №6.

Тема: Віртуальні функції і класи.

Мета: Отримати практичні навички по використанні віртуальних функцій.

Питання для обговорення: Динамічний поліморфізм. Таблиця віртуальних функцій. Віртуальний деструктор.

Література: 1, 3, 7.

Лабораторна робота №7.

Тема: Виключні ситуації.

Мета: Отримати практичні навички використання виключних ситуацій в C++.

Питання для обговорення: Перехоплення виключних ситуацій. Використання try, catch, throw. Обмеження виключних ситуацій.

Література: 10, 3.

Лабораторна робота №8.

Тема: Потоки.

Мета: Набути практичних навичок по використанні потоків.

Питання для обговорення: Робота з файлами. Перевантаження операцій роботи з потоками.
Література: 9, 1, 4.

Лабораторна робота №9.

Тема: Шаблони класів

Мета: Отримати практичні навички проектування шаблонів класів та елементарних контейнерів.

Питання для обговорення: Стек. Черга. Динамічні структури даних.

Література: 8, 10, 1.

Лабораторна робота №10.

Тема: Синхронізація між процесами та потоками.

Мета: Отримати практичні навички по створенню багатопотокового додатку.

Питання для обговорення: Threat. Дескриптори потоку. Література: 7, 8.

Лабораторна робота №11.

Тема: Робота з файлами з використанням класів.

Мета: Отримати практичні навички обміну даними з файловою системою.

Питання для обговорення: Модуль роботи з файловою системою. Обмін даними з файловою системою.

Література: 5.

Лабораторна робота №12.

Тема: Ієрархія класів.

Мета: Отримати практичні навички по проектуванні програмних додатків зі складною ієрархією класів.

Питання для обговорення: Ієрархія класів. Додатки зі складною ієрархією класів.

Література: 9, 6.

6. Комплексне практичне індивідуальне завдання з дисципліни.

Індивідуальна робота студента передбачає виконання комплексного практичного індивідуального завдання, яке виконується кожним студентом одноосібно за одним з варіантів:

- Теоретичний блок.
- Розробка програми для розв'язання прикладного завдання.
- Розробка програми для факторизації багато розрядних чисел.
- Реалізація алгоритму Евкліда.
- Китайська теорема про залишки.
- Генерація випадкової точки на еліптичній кривій.
- Ймовірнісний тест простоти числа.
- Кодування файлу обраним алгоритмом.
- Реалізація симетричного алгоритму шифрування.
- Реалізація алгоритму Гешування.

7. Самостійна робота

№ п/п	Тематика
1	Функціонал класів ostream і ios. Форматування виводу.
2	Абстракція даних.
3	Оператори управління МП.
4	Додавання до std::string.
5	Вивід об'єктів класів через оператор виводу.
6	Вбудовані функції.
7	Параметри за замовчуванням.
8	Стек і Купа.
9	Ємність вектора.
10	Рекурсія і Числа Фібоначчі.
11	Обробка помилок, cerr і exit().
12	assert і static assert.

13	Еліпсис.
14	Лямбда-вирази.
15	Лямбда-захоплення.
16	Контейнери вказівників на функції.
17	Sstring.
18	Обчислювальні структури послідовностей. Списки. Лінійні списки.
	Тренінг.
Разом:	

8. Тренінг з дисципліни.

Порядок проведення тренінгу:

1. Вступна частина проводиться з метою ознайомлення студентів з темою тренінгу.
2. Організаційна частина полягає у створенні робочого настрою у колективі студентів.
3. Практична частина реалізується шляхом виконання завдань з певних проблемних питань теми тренінгу.
4. Підведення підсумків. Обговорення результатів виконаних завдань. Обмін думками з питань, що виносились на тренінг.

№	Вид роботи	Порядок проведення тренінгу
1	Огляд сучасних середовищ розробки комп'ютерних програм	1. Розгляд сучасних середовищ розробки. 2. Дослідження бібліотек програмних функцій для створення програмних додатків по роботі з числовими та символічними змінними.
2	Розгляд процесу розробки алгоритму роботи програмних додатків на основі блок-схем	- Постановка задачі. - Опис технічного завдання. - Графічне відображення алгоритму роботи програми.
3	Реалізація розробленого алгоритму роботи програми на мові програмування C++	- Реалізація алгоритму роботи програми на мові програмування C++. - Реалізація інтерфейсу користувача.
4	Тестування розробленого програмного додатку	1. Вибір та обґрунтування тестової вибірки. 2. Опис вхідних та вихідних даних. 3. Перевірка правильності роботи реалізованого програмного додатку.

9. Методи навчання.

У навчальному процесі застосовуються: лекції, в тому числі з використання мультимедійного проектора та інших ТЗН; практичні роботи, індивідуальні заняття; робота в Інтернет.

10. Засоби оцінювання та методи демонстрування результатів навчання.

У процесі вивчення дисципліни „Програмування П” використовуються наступні засоби оцінювання та методи демонстрування результатів навчання:

- стандартизовані тести;
- поточне опитування;
- залікове модульне тестування та опитування;
- презентації результатів виконаних завдань;
- оцінювання результатів КПЗ;
- завдання на лабораторному обладнанні, тощо;
- ректорська контрольна робота;
- екзамен;
- інші види індивідуальних та групових завдань.

11. Критерії, форми поточного та підсумкового контролю

Підсумковий бал (за 100-бальною шкалою) з дисципліни „Програмування П” визначається як середньозважена величина, залежно від питомої ваги кожної складової залікового кредиту:

Заліковий модуль 1	Заліковий модуль 2 (ректорська контрольна робота)	Заліковий модуль 3 (підсумкова оцінка за КПЗ, враховуючи поточне опитування)	Заліковий модуль 4 (письмовий екзамен)
20 %	20 %	20 %	40 %
1. Теоретичні питання: 9 тем по 2 бали – мах 18 балів. 2. Письмова робота – мах 58 балів. 2. Практичне завдання: 6 практичних завдань по 4 бали – мах 24 бали.	1. Теоретичні питання: 9 тем по 2 бали – мах 18 балів. 2. Письмова робота – мах 58 балів. 2. Практичне завдання: 6 практичних завдань по 4 балів – мах 24 балів.	1. Підготовка КПЗ – мах 40 балів. 2. Захист КПЗ – мах 40 балів. 3. Участь у тренінгах – мах 20 балів	1. Теоретичні питання: 3 питання по 25 балів мах 75 балів. 2. Практичне завдання мах 25 балів

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)

12. Інструменти, обладнання та програмне забезпечення, використання яких передбачає

№	Найменування
1.	Обладнання: проектор, комп'ютери з доступом до мережі Інтернету.
2.	Програмне забезпечення: gcc, 7zip, DjVu Viewer, Visual studio community

РЕКОМЕНДОВАНІ ДЖЕРЕЛА ІНФОРМАЦІЇ

- Andrist B., Sehr V. C++ High Performance: Master the art of optimizing the functioning of your C++ code. 2nd edition. - Packt Publishing, 2021. - 540 p.
- Bancila M. Modern C++ Programming Cookbook. Packt, 2020. — 751 p.
- Deitel Paul, Deite Harveyl. C++20 for Programmers. 3rd Edition (Rough Cuts). - Pearson Education, 2021. — 2155 p.
- Bansal Paritosh. Computer Programming in C. Krishna Prakashan Media Pvt Ltd, 2021. - 622 p.
- Кедик Дмитро. Реалізація корисних алгоритмів у C++. Independent Published, 2020. - 710 c.
- BDM. C++ & Python Complete Manual. 11th Edition. - Papercut Limited, 2022. - 148 p.
- Kirk Dorothy R. Demystified Object-Oriented Programming with C++: Implement proven object-oriented design principles to write better code and create robust software. Packt Publishing, 2021. — 568 p.
- Bouras Aristides. C++ and Algorithmic Thinking for the Complete Beginner: Learn to Think Like a Programmer/2nd edition. - Independently published, 2021. - 1056 p.
- Briggs W. C++20 for Lazy Programmers: Quick, Easy, and Fun C++ for Beginners. 2nd.ed. - Apress, 2020. - 706p.
- Browning J. Burton, Sutherland Bruce. C++20 Recipes: A Problem-Solution Approach. 2nd edition. - Apress, 2020. - 657 p.
- Cyganek B. Introduction to Programming with C++ for Engineers. Hoboken: Wiley, 2021. – 649 p.
- Mueller J.P. C++ All-in-One For Dummies. 4th ed. - Wiley, 2021. — 915 p.
- Davidson J. Guy, Gregory Kate. Beautiful C++: 30 Core Guidelines for Writing Clean, Safe, and Fast Code (Final). Addison-Wesley Professional, 2022. - 352 p.
- Deitel P., Deitel H. C How to Program. 9th ed. - Pearson, 2022. - 838 p.
- Singer Adam B. Practical C++ Design: From Programming to Architecture. 2nd Edition. - Apress

Media LLC, 2022. - 303 p.

16. Fertig A. Programming with C++20: Concepts, Coroutines, Ranges, and more. Fertig Publications, 2021. - 344 p.

17. Gladstone Adam. C++ Software Interoperability for Windows Programmers. Apress Media LLC, 2022. - 235 p.