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

ЗАТВЕРДЖУЮ
Декан факультету комп'ютерних інформаційних технологій
_____Ігор ЯКИМЕНКО
«29» 08 _____ 20 25 р.

ЗАТВЕРДЖУЮ
Проректор з науково-педагогічної роботи
_____Віктор ОСТРОВЕРХОВ
«29» 08 _____ 20 25 р

РОБОЧА ПРОГРАМА
з дисципліни
«ПРОГРАМУВАННЯ ДЛЯ КІБЕРБЕЗПЕКИ»

Ступінь вищої освіти – **бакалавр**
Галузь знань – **F Інформаційні технології**
Спеціальність – **F5 Кібербезпека та захист інформації**
Освітньо-професійна програма – **Кібербезпека**

Кафедра кібербезпеки

Форма навчання	Курс	Семестр	Лекції (год.)	Лабор. (год.)	ІРС (год.)	Тренінг (год)	СРС (год.)	Разом (год.)	Екзамен (сем)
Денна	1	2	30	30	4	8	78	150	2

29.08.2025

Тернопіль – 2025

Робоча програма складена на основі освітньо-професійної програми «Кібербезпека» першого (бакалаврського) рівня вищої освіти за спеціальністю F5 Кібербезпека та захист інформації галузі знань F Інформаційні технології, затвердженої Вченою радою ЗУНУ, протокол №8 від 26.06.2025р.

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

Робоча програма затверджена на засіданні кафедри кібербезпеки,
протокол № 1 від 26.08.2025 р.

Завідувач кафедри
кібербезпеки Василь ЯЦКІВ

Гарант ОП Михайло КАСЯНЧУК

СТРУКТУРА РОБОЧОЇ ПРОГРАМИ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

1. Опис дисципліни „ Програмування для кібербезпеки **”**

Дисципліна – «Програмування для кібербезпеки»	**Галузь знань, спеціальність, СВО**	**Характеристика навчальної дисципліни**
Кількість кредитів – 5	Галузь знань - F - «Інформаційні технології»	Статус дисципліни – обов’язкова Мова навчання - українська
Кількість залікових модулів – 5	Спеціальність – F5 - «Кібербезпека та захист інформації»	Рік підготовки – 1 Семестр – 2
Кількість змістових модулів –3	Ступінь вищої освіти – бакалавр	Лекції: 30 год. Лабораторні заняття: 30 год.
Загальна кількість годин – 150		Самостійна робота: 78 год. Тренінг – 8 год. Індивідуальна робота -4 год.
Тижневих годин: 10 год., з них аудиторних – 4 год.		Вид підсумкового контролю – екзамен

2. Мета й завдання вивчення дисципліни «Програмування для кібербезпеки**»**

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 технології.

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 С++. 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 вивченя дисципліни

ЗКІ. Здатність застосовувати знання у практичних ситуаціях.

СК2. Здатність використовувати інформаційні технології, сучасні методи і моделі кібербезпеки та системи захисту інформації.

СК 4. Здатність забезпечувати захист інформації в інформаційних та інформаційно-комунікаційних системах згідно встановленої політики кібербезпеки й захисту інформації.

СК10. Здатність виконувати моніторинг інформаційних процесів, аналізувати, виявляти, оцінювати можливі вразливості та загрози інформаційному простору й інформаційним ресурсам згідно з встановленою політикою інформаційної безпеки.

2.4 Передумови для вивчення дисципліни.

Вивчення курсу «Програмування для кібербезпеки» передбачає наявність систематичних та ґрунтовних знань із суміжних курсів «Основи кібербезпеки», «Основи програмування, а також цілеспрямованої роботи на лекційних та лабораторних заняттях, самостійної роботи.

2.5. Результати навчання

РН4. Організовувати власну професійну діяльність, обирати й використовувати оптимальні методи та способи розв'язання складних спеціалізованих задач і практичних проблем у професійній діяльності, оцінювати їхню ефективність.

РН5. Аналізувати, аргументувати, приймати рішення при розв'язанні складних спеціалізованих задач і практичних завдань у професійній діяльності, які характеризуються комплексністю та неповною визначеністю умов, відповідати за прийняті рішення.

РН7. Застосовувати й адаптувати теорії інформації та кодування, математичної статистики, чисел, криптографії та стеганографії, оброблення і передачі сигналів тощо, принципи, методи, поняття кібербезпеки та захисту інформації у навчанні та професійній діяльності.

РН8. Застосовувати знання й розуміння математики та фізики в професійній діяльності, формалізувати задачі предметної галузі кібербезпеки та захисту інформації, формулювати їх математичну постановку та обирати раціональний метод вирішення.

РН10. Використовувати сучасні інформаційні технології, методи і моделі кібербезпеки та систем захисту інформації для здійснення професійної діяльності.

3. Програма навчальної дисципліни «Програмування для кібербезпеки»

Змістовий модуль 1. Основи об'єктно орієнтованого програмування.

Тема 1. Основи ООП в С++.

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

Тема 2. Конструктори.

1. Список ініціалізації членів класу. 2. Ініціалізація нестатичних членів класу. 3. Делегуючі конструктори. 4. Деструктори. 5. Прихований вказівник *this. 6. Анонімні об'єкти. 7. Вкладені типи даних в класах. 8. Вимірювання часу виконання (таймінг) коду.

Тема 3. Можливості використання класів.

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

Тема 4. Перевантаження операторів в С++.

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.

Тема 5. Конструктор копіювання

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

Змістовий модуль 2. Ієрархія класів.

Тема 6. Зв'язки між об'єктами в С++.

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.

Тема 7. Спадкування в С++.

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

Тема 8. 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.

Тема 9. Віртуальні функції в С++.

1. Віртуальні функції в С++. 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.

Тема 10. Чисті віртуальні функції.

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

Тема 11. Динамічне приведення типів.

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

Тема 12. Шаблони в С++.

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

Змістовий модуль 3. Спеціальний розділ.

Тема 13. Винятки в С++.

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.

Тема 14. Розумні вказівники і Семантика переміщення в С++

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

Тема 15. 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. 8. Контейнери STL. 9. Ітератори STL. 10. Алгоритми STL. 11. Рядкові класи std::string і std::wstring. 12. Створення, знищення і конвертація std::string. 13. Довжина і ємність std::string. 44. Доступ до символів std::string. Конвертація std::string в рядки C-style.

4. Структура залікового кредиту дисципліни

	Кількість годин					
	Лекції	Лабор. заняття	СРС	ІРС	Тренінг	Контрольні заходи
Змістовий модуль 1 Основи об'єктно орієнтованого програмування						
1. Основи ООП в С++.	2	2	5	1	2	Поточне опитування
2. Конструктори.	2	2	5			
3. Можливості використання класів.	2	2	5			
4. Перевантаження операторів в С++.	2	2	5			
5. Конструктор копіювання.	2	2	5			
Змістовий модуль 2. Ієрархія класів						
6. Зв'язки між об'єктами в С++.	2	2	5	2	4	Поточне опитування
7. Спадкування в С++.	2	2	5			
8. Спадкування і специфікатори доступу.	2	2	5			
9. Віртуальні функції в С++.	2	2	5			
10. Чисті віртуальні функції.	2	2	5			
11. Динамічне приведення типів.	2	2	5			
12. Шаблони в С++.	2	2	5			
Змістовий модуль 3. Спеціальний розділ						
13. Винятки в С++.	2	2	6	1	2	Поточне опитування
14. Розумні вказівники.Семантика переміщення в С++.	2	2	6			
15. Потоки вводу і виводу.	2	2	6			
Разом	**30**	**30**	**78**	**4**	**8**	**Екзамен**

5. Тематика лабораторних робіт.

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

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.

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.

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.

Лабораторна робота №5.

Тема: Успадкування класів.
Мета: Отримати практичні навички по реалізації успадкування.
Питання для обговорення: Успадкування, Множинне спадкування. Використання віртуальних функцій.

Лабораторна робота №6.

Тема: Віртуальні функції і класи.
Мета: Отримати практичні навики по використанні віртуальних функцій.
Питання для обговорення: Динамічний поліморфізм. Таблиця віртуальних функцій. Віртуальний деструктор.

Лабораторна робота №7.

Тема: Виключні ситуації.
Мета: Отримати практичні навики використання виключних ситуацій в С++.
Питання для обговорення: Перехоплення виключних ситуацій. Використання try, catch, throw. Обмеження виключних ситуацій.

Лабораторна робота №8.

Тема: Потоки.
Мета: Набути практичних навичок по використанні потоків.
Питання для обговорення: Робота з файлами. Перевантаження операцій роботи з потоками.

Лабораторна робота №9.

Тема: Шаблони класів
Мета: Отримати практичні навички проектування шаблонів класів та елементарних контейнерів.
Питання для обговорення: Стек. Черга. Динамічні структури даних.

Лабораторна робота №10.

Тема: Синхронізація між процесами та потоками.
Мета: Отримати практичні навички по створенню багатопотокового додатку.
Питання для обговорення: Threat. Дескриптори потоку.

Лабораторна робота №11.

Тема: Робота з файлами з використанням класів.
Мета: Отримати практичні навики обміну даними з файловою системою.
Питання для обговорення: Модуль роботи з файловою системою. Обмін даними з файловою системою.

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.

6. Організація та проведення тренінгу з курсу.

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 написання програмного засобу та алгоритму його роботи за обраною темою.
4. Підведення підсумків. Обговорення результатів виконаних завдань. Обмін думками з

питань, що виносились на тренінг.

Тематика тренінгу: Сучасні середовища розробки комп'ютерних програм.

Мета тренінгу: забезпечення студентів теоретичними знаннями та практичними навичками процесу розробки алгоритму роботи програмних додатків на основі блок-схем.

Здобувачі обирають одну із запропонованих тем або формулюють власну тему дослідження у межах проблематики.

Завдання тренінгу: Реалізація розробленого алгоритму роботи програми на мові програмування С++.

Орієнтовна тематика:

– 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.

Самостійна робота з курсу «Аналіз шкідливого програмного забезпечення» включає вивчення наведених тем та оцінюється у вигляді тесту:

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.
19. Списки.
20. Лінійні списки.

8. Методи навчання.

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

9. Засоби оцінювання та методи демонстрування результатів навчання.

У процесі вивчення дисципліни „ Програмування для кібербезпеки " використовуються наступні засоби оцінювання та методи демонстрування результатів навчання:

- поточне тестування та опитування;
- підсумкова письмова робота за темами;
- презентації результатів виконаних завдань;
- оцінювання виконання лабораторних робіт;;
- оцінювання тренінгів;
- оцінювання результатів самостійної роботи
- підсумковий екзамен.

10. Політика оцінювання

- *Політика щодо дедлайнів і перескладання*. Для виконання усіх видів завдань здобувачами освіти і проведення контрольних заходів встановлюються конкретні терміни. Перескладання модулів проводиться в установленому порядку.
- *Політика щодо академічної доброчесності*. Списування під час проведення контрольних заходів заборонені. Під час контрольного заходу учасник може користуватися лише дозволеними допоміжними матеріалами або засобами, йому забороняється в будь-якій формі обмінюватися інформацією з іншими учасниками, використовувати, розповсюджувати, збирати варіанти контрольних завдань.
- *Політика щодо відвідування*. За об'єктивних причин (наприклад, карантин, воєнний стан, хвороба, закордонне стажування) навчання може відбуватись в дистанційній формі за погодженням із керівником курсу з дозволу дирекції факультету.

11. Критерії, форми поточного та підсумкового контролю

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

Модуль 1		Модуль 2		Модуль 3	Модуль 4	Модуль 5
10%	10%	10%	10%	5%	15%	40%
Поточне оцінювання	Модульний контроль 1	Поточне оцінювання	Модульний контроль 2	Тренінги	Самостійна робота	Екзамен
Оцінка за даний модуль визначається як середнє арифметичне за захист лабораторних робіт №1-6.	Підсумкове модульне тестування за темами №1-8.	Оцінка за даний модуль визначається як середнє арифметичне за захист лабораторних робіт №7-12.	Підсумкове модульне тестування за темами № 9-15.	Визначається як середнє арифметичне з оцінок за виконання та презентацію одного завдання тренінгу.	Підсумкове тестування за результатами опрацювання теоретичного матеріалу. Теоретичні питання: 25 питань по 4 бали - max 100 балів	Теоретичні питання: у вигляді тестів - max 60 балів. Практичне завдання - max 40 балів

Виконання лабораторних робіт:

90-100 балів (Відмінно) - здобувач освіти самостійно, без помилок, виконав усі кроки в рамках лабораторної роботи, правильно задокументував етапи та вільно оперує поняттями та принципами, що відносяться до теми дисципліни.

75-89 балів (Добре) - здобувач освіти виконав завдання лабораторної роботи, проте в процесі виконання допустив кілька дрібних помилок, які не вплинули на кінцевий результат (наприклад, неточна послідовність дій), в процесі роботи виникали додаткові запитання.

60-74 балів (Задовільно) - здобувач освіти виконав завдання лабораторної роботи, але з суттєвими помилками, наприклад, не з першого разу чи не до кінця. Розуміння поставлених у лабораторній роботі завдань є поверхневим та неповним.

1-59 балів (Незадовільно) - здобувач освіти не зміг виконати завдання або результати були повністю невірними. Не продемонстрував базових навичок роботи з програмним забезпеченням.

Підсумкове модульне тестування:

90–100 балів – правильно виконано 90–100% завдань тесту.

75–89 балів – правильно виконано 75–89% завдань тесту.

60–74 бали – правильно виконано 60–74% завдань тесту.

1–59 балів – правильно виконано менше 60% завдань тесту.

Тренінг:

90-100 балів (Відмінно) - здобувач освіти самостійно, без помилок, виконав усі етапи завдання, правильно задокументував усі етапи роботи, та вільно оперує поняттями та принципами дисципліни.

75-89 балів (Добре) - здобувач освіти виконав завдання, але з кількома дрібними помилками, які не вплинули на кінцевий результат, в процесі роботи виникали додаткові запитання.

60-74 балів (Задовільно) - здобувач освіти виконав завдання, але з суттєвими помилками, наприклад, не з першого разу. Розуміння поставлених у тренінгу завдань є поверхневим.

1-59 балів (Незадовільно) - здобувач освіти не зміг виконати завдання або результати були повністю невірними. Не продемонстрував достатній рівень навичок роботи з апаратним та програмним забезпеченням.

Самостійна робота: оцінюється у вигляді тестування за результатами опрацювання теоретичного матеріалу. Теоретичні питання: 25 питань по 4 бали, Максимальна оцінка 100 балів.

Підсумковий екзамен - вид підсумкового контролю, який проводиться з метою оцінювання засвоєння здобувачем вищої освіти теоретичного та практичного матеріалу з дисципліни «Програмування для кібербезпеки». Екзаменаційний білет складається з двох блоків.

Перший блок містить теоретичні питання і вигляді тестів, за кожну правильну відповідь тестування здобувач освіти отримує 4 бали, максимум 60 балів. Виконання практичних завдань передбачається у текстовому вигляді (засобами системи дистанційного навчання) та можуть бути прокоментовані усно для пояснення шляху реалізації поставленого практичного завдання. Максимальна кількість балів за виконання практичного завдання 40 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

1. Куліков В.М., Рябцев В.В., Паршуков С.С. Об’єктно-орієнтоване програмування для фахівців з кібербезпеки: навч. посіб. / ІСЗЗІ КПІ ім. Ігоря Сікорського. Київ: КПІ ім. Ігоря Сікорського, 2023. 365 с.
2. Бурячок В. Л. Основи інформаційної та кібернетичної безпеки. [Навчальний посібник]. / В. Л. Бурячок , Р. В. Киричок, П. М. Складанний – К. , 2018. – 320 с.
3. Вступ до програмування мовою С++. Організація обчислень : навч. посіб. / Ю. А. Бєлов, Т. О. Карнаух, Ю. В. Коваль, А. Б. Ставровський. – К. : Видавничо-поліграфічний центр "Київський університет", 2012. – 175 с.
4. 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. – 477 с.
5. «Об'єктно-орієнтоване програмування. Практикум»: навч. посіб. для студентів спеціальності 163 «Біомедична інженерія» / КПІ ім. Ігоря Сікорського; уклад. В.А. Данілова.– КПІ ім. Ігоря Сікорського.- 2021. – 121 с
6. BDM. C++ & Python Complete Manual. 11th Edition. - Papercut Limited, 2022. - 148 p.

7. 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.
8. Bouras Aristides. C++ and Algorithmic Thinking for the Complete Beginner: Learn to Think Like a Programmer/2nd edition. - Independently published, 2021. - 1056 p.
9. Briggs W. C++20 for Lazy Programmers: Quick, Easy, and Fun C++ for Beginners. 2nd.ed. - Apress, 2020. - 706p.
10. Browning J. Burton, Sutherland Bruce. C++20 Recipes: A Problem-Solution Approach. 2nd edition. - Apress, 2020. - 657 p.
11. Cyganek B. Introduction to Programming with C++ for Engineers. Hoboken: Wiley, 2021. – 649 p.
12. Mueller J.P. C++ All-in-One For Dummies. 4th ed. - Wiley, 2021. — 915 p.
13. Davidson J. Guy, Gregory Kate. Beautiful C++: 30 Core Guidelines for Writing Clean, Safe, and Fast Code (Final). Addison-Wesley Professional, 2022. - 352 p.
14. Deitel P., Deitel H. C How to Program. 9th ed. - Pearson, 2022. - 838 p.
15. 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: Conceptes, Coroutines, Ranges, and more. Fertig Publications, 2021. - 344 p.
17. Gladstone Adam. C++ Software Interoperability for Windows Programmers. Apress Media LLC, 2022. - 235 p.