

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ  
ІМЕНІ СЕМЕНА КУЗНЕЦЯ

**ЗАТВЕРДЖУЮ"**  
Заступник керівника  
(проректор з науково-педагогічної роботи)  
  
М. В. Афанасьєв

**"ПРОГРАМУВАННЯ"**  
робоча програма навчальної дисципліни

Галузь знань **12 "ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ"**  
Спеціальність **125 "КІБЕРБЕЗПЕКА"**  
Освітній рівень **перший (бакалаврський)**  
Освітня програма **"КІБЕРБЕЗПЕКА"**

Вид дисципліни **базова**  
Мова викладання, навчання та оцінювання **українська**

Завідувач кафедри кібербезпеки та інформаційних технологій  **Євсєєв С.П.**

Харків  
ХНЕУ ім. С. Кузнеця  
2019

ЗАТВЕРДЖЕНО  
на засіданні кафедри кібербезпеки  
та інформаційних технологій  
Протокол № 1 від 26.08.2019 р.

Розробник(-и):  
Мілов О.В., к.т.н., доцент кафедри КІТ

**Лист оновлення та перезатвердження  
робочої програми навчальної дисципліни**

Навчальний рік	Дата засідання кафедри – розробника РПНД	Номер протоколу	Підпис завідувача кафедри

## 1. Вступ

### Анотація навчальної дисципліни:

Подано тематичний план навчальної дисципліни й її змістовність за модулями та темами, вміщено плани лекцій і лабораторних занять, матеріал щодо закріплення знань (завдання для самостійної роботи, контрольні запитання), методичні рекомендації та оцінювання знань студентів.

Дисципліну "Програмування" віднесено до групи освітньо-професійних дисциплін підготовки бакалаврів за спеціальністю 125 "Кібербезпека".

Сьогоднішні умови господарювання вимагають від фахівців з економічного управління всебічного використання новітніх інформаційних технологій. Широкі можливості комп'ютеризованих засобів в питаннях збирання, оброблення та видавання необхідної інформації здатні значно підвищити якість економічних розрахунків, зробити більш ефективним процес обґрунтування економічних рішень.

Навчальна дисципліна "Програмування" є інструментальною основою для виконання аналітичної частини подальших спецкурсів, а також курсових і дипломних робіт. Вона забезпечує такі дисципліни: "Об'єктно-орієнтоване програмування", "Алгоритми та структури даних", "Операційні системи", "Бази даних", "Розподілені та паралельні обчислення", "WEB-технології та WEB-дизайн", "Технології розробки та тестування програмного забезпечення", "Кросплатформне програмування", "Програмування для мобільних пристроїв".

Навчальна дисципліна "Програмування" є базовою навчальною дисципліною та вивчається згідно з навчальним планом підготовки фахівців освітнього ступеня "бакалавр" за спеціальністю 125 "Кібербезпека" усіх форм навчання.

### Мета навчальної дисципліни:

**Метою** викладання цієї навчальної дисципліни є засвоєння необхідних знань щодо основних понять алгоритмізації і техніки застосування у програмуванні базових алгоритмічних структур (організація програм) і базових типів даних. При цьому велика увага приділяється практичній роботі студентів на персональних комп'ютерах.

Курс	1	
Семестр	1	
Кількість кредитів ECTS	5	
Аудиторні навчальні заняття	лекції	32
	семінарські, практичні	–
	лабораторні	32
Самостійна робота	86	
Форма підсумкового контролю	екзамен	

### Структурно-логічна схема вивчення навчальної дисципліни:

Попередні дисципліни	Наступні дисципліни
Вища математика	Основи криптографічного захисту
Методи та засоби комп'ютерних інформаційних технологій	Основи технічного захисту інформації
Технології обробки інформації	Забезпечення інформаційної безпеки

## 2. Компетентності та результати навчання задисципліною:

Компетентності	Результати навчання
Приймати оптимальні рішення щодо розроблення програмних засобів.	Визначати цільові програмні платформи Визначати інструментальні засоби розроблення програмних засобів. Визначати оптимальні методи розроблення програмних засобів .
Розробляти ефективні алгоритми та програми з використанням математичних методів та моделей	Визначати способи формалізації та реалізації програмних засобів. Здійснювати обґрунтований вибір математичних методів та моделей для їх програмної реалізації. Здійснювати розроблення алгоритмів реалізації математичних методів та моделей. Здійснювати кодування розроблених алгоритмів на мові програмування високого рівня. Застосовувати методи відлагодження та тестування програмних засобів. Здійснювати оцінювання отриманих результатів.
Володіти культурою мислення, здатним до узагальнення, аналізу, сприйняття інформації, постановці мети і вибору шляхів її досягнення	Знати основні типи дефектів програм і методи налагодження програм Володіти навичками виявлення та усунення дефектів у своїх програмах на мові Сі
Освоювати методики використання програмних засобів для вирішення практичних завдань	Знати основні структури даних, алгоритми розв'язання типових задач програмування Вміти реалізовувати основні структури даних і алгоритми розв'язання типових задач на мові Сі Володіти навичками використання мови Сі і стандартних бібліотек мови Сі

## 3. Програма навчальної дисципліни

### Змістовий модуль 1. Організація програм

#### Тема 1. Етапи розроблення та впровадження програм. Поняття алгоритму та типові алгоритмічні структури програмування

Поняття алгоритму. Властивості алгоритму. Типові алгоритмічні конструкції. Розроблення алгоритму методом покрокового уточнення. Алгоритмічні конструкції: послідовність, вибір, повторення. Способи завдання алгоритму. Критерії оцінювання алгоритмів.

Початкові відомості про технологію програмування: процедурне, структурне і об'єктно-орієнтоване програмування. Мови програмування: процедурні, аплікативні, системи правил, об'єктно-орієнтовані. Стандартизація мов та середовища проектування. Транслятори. Редактори. Компонувальники. Відладчики. Керівні структури: оператори, вирази та підпрограми. Огляд сучасних інтегрованих систем програмування. Інтегроване середовище системи програмування TurboC. Етапи розроблення та впровадження програм. Вимоги до програмного коду.

## **Тема 2. Архітектура комп'ютерів, принципи Джона фон Неймана**

Електронний числовий інтегратор ENIAC. Логічна організація універсального обчислювального пристрою. Відкрита архітектура системних блоків. Принципи Джона фон Неймана.

## **Тема 3. Позиційні системи числення**

Класифікація систем числення. Особливості позиційних систем числення. Основа, алфавіт та базис позиційних систем числення. Подання чисел в позиційних системах числення. Переведення чисел з однієї системи числення в іншу.

## **Тема 4. Елементи алгоритмічних мов: концепція типів даних, імена, значення, покажчики, змінні, константи, операції, вирази**

Стандарти мов програмування C. Структура C програми.

Лексичні елементи мови C: алфавіт, коментарі, ідентифікатори, службові слова, дані, вираз, операнд, змінна, операція. Домовленості про імена.

Поняття типу даних. Класифікація і подання даних. Базові типи даних: логічний, символний, цілий, речовинний. Перетворення типів: неявні перетворення, явні перетворення типу.

Пріоритети операцій. Таблиця пріоритетності й асоціативності операцій.

Операції. Унарні операції: унарний мінус, унарний плюс, порозрядне інвертування, логічне заперечення, інкремент, декремент, операція обчислення розміру (зігеої). Бінарні операції: адитивні, мультиплікативні, операції зсувів, порозрядні, операції відносин, логічні, привласнення, операція "кома".

Стандартні математичні функції.

Константні величини: цілі, дійсні, перелічувальні, символні (літерні), рядкові (рядки або літерні рядки). Правила визначення компілятором констант. Визначення констант за допомогою ключового слова `const`.

## **Тема 5. Структурне програмування: послідовність, розгалуження та цикли**

Загальні відомості про систему вводу-виводу даних.

Вирази, символи пропусків, блоки і комплексні вирази. Операнд, змінна. Оператор привласнення. Оголошення й ініціалізація змінних.

Типи операторів. Найпростіший оператор, оператор-оголошення, оператор-визначення, оператор-вираз.

Керівні оператори: оператори проходження; оператори вибору (єдиний вибір - `if`, подвійний вибір - `if/else`, множинний вибір - `switch`, умовна операція); оператори повторення (оператор `while`, оператор `for`). Вкладені цикли. Керівні оператори в циклах: оператор `break`, оператор `continue`, оператор `goto`. Рекомендації щодо вибору циклів.

## **Тема 6. Перед процесорна обробка**

Директиви препроцесора. Основи апарату макросів.

Директива препроцесора `#include` і файли, що включаються. Директива препроцесора `#define`: оголошення констант і макросів. Умовна компіляція. Використовування ключового слова `typedef`. Відмінність директиви `#define` від оператора `typeid`.

## **Тема 7. Процедурно-орієнтоване програмування. Рекурсія**

Загальні відомості про функції. Структура функції.

Значення, параметри і аргументи, що повертаються. Оголошення функції. Прототипи функцій. Визначення функції. Виконання функції.

Локальні та глобальні змінні. Правило видимості змінних. Перетворення типів аргументів функцій. Правила автоматичного (неявного) перетворення типів. Явні перетворення типів. Правила роботи з функціями. Класи пам'яті.

Список параметрів функції. Параметри за умовчанням. Способи передачі параметрів. Способи повернення значення.

Функції, що підставляються.

Створення власних заголовних файлів.

Перевантаження функцій. Рекурсія. Типи рекурсії.  
Робота функцій. Розбиття пам'яті. Стек і функції.  
Модифікатори функцій.

### **Тема 8. Бібліотеки динамічного компонування dll**

Статичні й динамічні бібліотеки. Загальні відомості про DLL. Структура DLL  
Оголошення функції в DLL.

Способи завантаження DLL. Явне (статичне) завантаження DLL. Неявне  
(динамічне) завантаження DLL.

### **Тема 9. Методології розроблення програм: низхідне та висхідне проектування, модульне програмування**

Методологія розроблення ПЗ: каскадне розроблення, інтерактивне розроблення.  
Проектування програм, принцип модульності.

Методи проектування програм: низхідне та висхідне проектування, метод розширення ядра.

## **Змістовий модуль 2. Основні похідні типи даних мов програмування C**

### **Тема 10. Масиви**

Оголошення масивів. Ініціалізація масивів. Обробка одновимірних масивів даних економічного характеру.

Алгоритми сортування масивів.

Багатовимірні масиви. Ініціалізація багатовимірного масиву. Типові приклади обробки матриць.

Масиви і графи.

Масиви як параметри функцій.

### **Тема 11. Похідні типи даних. Рядки в стилі C**

Рядки як масиви символів. Операції з рядками. Ввід-вивід рядків. Поняття покажчика, посилання. Покажчики і масиви. Адресна арифметика. Посилання. Приклади використання покажчиків і посилань. Покажчики на функції.

Параметри функцій як посилання.

Організація пам'яті в сучасних процесорах і покажчики мови C. Моделі пам'яті. Статичні й динамічні змінні. Оператори `sizeof` і `sizeof`. Динамічні масиви. Динамічні масиви як параметри функцій.

### **Тема 12. Введення в систему вводу-виводу C**

Базові положення системи вводу-виводу C. Потоки і буфери. Стандартні об'єкти вводу-виводу.

Ввід даних за допомогою глобального об'єкта `cin`.

Вивід рядків. Введення одного символу. Використовування функції `get()`: без параметрів, з параметрами. Ввід рядків із стандартного пристрою введення. Використовування функції `getline()`. Вивід даних за допомогою глобального об'єкта `cout`. Очищення буфера виводу - `flush()`. Використовування функцій `put()` і `write()`. Ввід - вивід даних, що форматуються. Маніпулятори вводу-виводу. Функції `width()`, `precision()`, `fill()`. Маніпулятори, визначувані користувачем.

### **Теми лабораторних робіт**

Лабораторна робота №1 . Основні концепти програмування на прикладі мови Scratch.

Лабораторна робота №2. Робота з циклами та змінними в мові програмування C.

Лабораторна робота №3. Використання масивів та типу даних `char`. в C Робота з аргументами командного рядка.

Лабораторна робота №4. Особливості роботи з файлами в C.

Лабораторна робота №5. Реалізація структур даних та зв'язаних списків в C.

#### 4. Порядок оцінювання результатів навчання

Система оцінювання сформованих компетентностей у студентів враховує види занять, які згідно з програмою навчальної дисципліни передбачають лекційні та лабораторні заняття, а також виконання самостійної роботи. Оцінювання сформованих компетентностей у студентів здійснюється за накопичувальною 100-бальною системою. Відповідно до Тимчасового положення "Про порядок оцінювання результатів навчання студентів за накопичувальною бально-рейтинговою системою" ХНЕУ ім. С. Кузнеця, контрольні заходи включають:

поточний контроль, що здійснюється протягом семестру під час проведення лекційних та лабораторних занять і оцінюється сумою набраних балів (максимальна сума – 60 балів; мінімальна сума, що дозволяє студенту скласти іспит, – 35 балів);

модульний контроль, що проводиться у формі колоквиуму як проміжний міні-екзамен з ініціативи викладача з урахуванням поточного контролю за відповідний змістовий модуль і має на меті *інтегровану* оцінку результатів навчання студента після вивчення матеріалу з логічно завершеної частини дисципліни – змістового модуля;

підсумковий/семестровий контроль, що проводиться у формі семестрового екзамену, відповідно до графіку навчального процесу.

Порядок проведення поточного оцінювання знань студентів. Оцінювання знань студента під час лабораторних занять та виконання індивідуальних завдань проводиться за такими критеріями:

розуміння, ступінь засвоєння теорії та методології проблем, що розглядаються; ступінь засвоєння фактичного матеріалу навчальної дисципліни; ознайомлення з рекомендованою літературою, а також із сучасною літературою з питань, що розглядаються; вміння поєднувати теорію з практикою при розгляді виробничих ситуацій, розв'язанні задач, проведенні розрахунків у процесі виконання індивідуальних завдань та завдань, винесених на розгляд в аудиторії; логіка, структура, стиль викладу матеріалу в письмових роботах і при виступах в аудиторії, вміння обґрунтовувати свою позицію, здійснювати узагальнення інформації та робити висновки; арифметична правильність виконання індивідуального та комплексного розрахункового завдання; здатність проводити критичну та незалежну оцінку певних проблемних питань; вміння пояснювати альтернативні погляди та наявність власної точки зору, позиції на певне проблемне питання; застосування аналітичних підходів; якість і чіткість викладення міркувань; логіка, структуризація та обґрунтованість висновків щодо конкретної проблеми; самостійність виконання роботи; грамотність подачі матеріалу; використання методів порівняння, узагальнення понять та явищ; оформлення роботи.

Загальними критеріями, за якими здійснюється оцінювання позааудиторної самостійної роботи студентів, є: глибина і міцність знань, рівень мислення, вміння систематизувати знання за окремими темами, вміння робити обґрунтовані висновки, володіння категорійним апаратом, навички і прийоми виконання практичних завдань, вміння знаходити необхідну інформацію, здійснювати її систематизацію та обробку, самореалізація на лабораторних заняттях.

Студента слід **вважати атестованим**, якщо сума балів, одержаних за результатами підсумкової/семестрової перевірки успішності, дорівнює або перевищує 60. Мінімумально можлива кількість балів за поточний і модульний контроль упродовж семестру – 35 та мінімумально можлива кількість балів, набраних на екзамені, – 25.

Підсумкова оцінка з навчальної дисципліни розраховується з урахуванням балів, отриманих під час заліку, та балів, отриманих під час поточного контролю за накопичувальною системою. Сумарний результат у балах за семестр складає: "60 і більше балів – зараховано", "59 і менше балів – не зараховано" та заноситься у залікову "Відомість обліку успішності" навчальної дисципліни.

### Розподіл балів за тижнями

(вказати засоби оцінювання згідно з технологічною картою)

Теми змістового модуля		Лекційні заняття	Захист лабораторних робіт	Поточні КР	Усього	
Змістовий модуль 1	Тема 1	1 тиждень	1	4	-	5
	Тема 2	2 тиждень	1	4	-	5
	Тема 3	3 тиждень	1	4	-	5
	Тема 4	4 тиждень	1	4	-	5
	Тема 5	5 тиждень	1	4	-	5
	Тема 6	6 тиждень	1	4	-	5
	Тема 7	7 тиждень	1	4	-	5
	Тема 8	8 тиждень	1	4	10	15
Змістовий модуль 2	Тема 9	9 тиждень	1	4	-	5
	Тема 10	10 тиждень	1	4	-	5
	Тема 10	11 тиждень	1	4	-	5
	Тема 11	12 тиждень	1	4	-	5
	Тема 11	13 тиждень	1	4	-	5
	Тема 12	14 тиждень	1	4	-	5
	Тема 12	15 тиждень	1	4	15	20
	Залік					
Усього			15	60	25	100

### Шкала оцінювання: національна та ЄКТС

Сума балів за всі види навчальної діяльності	Оцінка ЄКТС	Оцінка за національною шкалою	
		для екзамену, курсового проекту (роботи), практики	для заліку
90 – 100	A	відмінно	зараховано
82 – 89	B	добре	
74 – 81	C		
64 – 73	D		
60 – 63	E	задовільно	не зараховано
35 – 59	FX	незадовільно	
1 – 34	F		

## 5. Рекомендована література

### 5.1. Основна

1. Аммерааль Л. STL для програмистов на C++ / Л. Аммерааль ; пер. с англ. - Москва : ДМК, 1999. - 240 с. : ил.
2. Вирт Н. Алгоритмы и структуры данных / Н. Вирт ; пер. с англ. - Москва : Мир, 1989. - 360 с. : ил.
3. Дейтел Х. Как программировать на C++ / Х. Дейтел, П. Дейтел ; пер. с англ. -



- Москва : ЗАО "Издательство БИНОМ", 2008. - 1455 с.
4. Ишкова Э. А. С++. Начала программирования / Э. А. Ишкова. - Москва : ООО "БИНОМ-Пресс", 2004. - 368 с. : ил.
  5. Кнут Д. Искусство программирования для ЭВМ / Д. Кнут. - Москва : Мир.  
Т. 1 - 1997.  
Т. 2 - 1997.  
Т. 3 - 1998.
  6. Пирогов В. Ю. Программирование на Visual C++.NET / В. Ю. Пирогов. - Санкт-Петербург: БХВ-Петербург, 2003. - 800 с.
  7. Ритчи Д. М. Язык программирования С / Д. М. Ритчи, Брайан У. Керниган ; пер. с англ. - Москва : Издательский дом "Вильямс", 2009. - 304 с.
  8. Румянцев П. В. Азбука программирования в Win32 API / П. В. Румянцев. - Москва : Горячая линия - Телеком, 2001. - 312 с.
  9. Страуструп Б. Программирование: принципы и практика использования С++ / Б. Страуструп ; пер. с англ. - Москва : Издательский дом "Вильямс", 2011. - 1248 с.
  10. Харт Дж. М. Системное программирование в среде Windows / Джонсон М. Харт ; пер. с англ. - 3-е изд. - Москва : Издательский дом "Вильямс", 2005. - 592 с. : ил.
  11. Хортон А. Visual C++ 2010: полный курс / А. Хортон ; пер. с англ. - Москва : Издательский дом "Вильямс", 2011. - 1216 с. : ил.
  12. Щупак Ю. А. Эффективная разработка приложений / Ю. А. Щупак. - Санкт-Петербург : Питер, 2007. - 572 с. : ил.

## 5.2. Додаткова

13. Верма Р. Д. Справочник по функциям Win32 API / Р. Д. Верма. - Москва : Горячая линия - Телеком, 2002. - 488 с.
14. Марченко А. Л. С++. Бархатный путь / А. Л. Марченко. - Москва : Горячая линия - Телеком, 1999. - 400 с.
15. Павловская Т. А. С/С++. Программирование на языке высокого уровня / Т. А. Павловская. - Санкт-Петербург : Питер, 2006. - 461 с. : ил.
16. Подбельский В. В. Программирование на языке С / В. В. Подбельский, С. С. Фомин. - 2-е доп. изд. - Москва : Финансы и статистика, 2004. - 600 с.
17. Подбельский В. В. Язык С++ : учеб. пособ. / В. В. Подбельский. - 4-е изд. - Москва : Финансы и статистика, 1999. - 560 с.
18. Саймон Р. Windows 2000 API. Энциклопедия программиста / Р. Саймон ; пер. с англ. - Киев : ООО "ДиасофтЮП", 2002. - 1088 с.
19. Страуструп Б. Дизайн и эволюция языка С++ / Б. Страуструп ; пер. с англ. - Москва : ДМК, 2000. - 444 с.
20. Страуструп Б. Язык программирования С++. Специальное издание / Б. Страуструп ; пер. с англ. - 3-е изд. - Санкт-Петербург ; Москва : "Невский диалект" ; Изд. "БИНОМ", 2004. - 1104 с. : ил.
21. Шеферд Д. Программирование на Microsoft Visual C++.NET / Д. Шеферд ; пер. с англ. - Москва : Издательско-торговый дом "Русская редакция", 2003. - 928 с. : ил.

## 5.3. Інформаційні ресурси

22. Главная страница MSDN, MSDN по-русски [Электронный ресурс]. - Режим доступа : <http://msdn.microsoft.com/ru-ru/default.aspx>.
23. Интернет Университет Информационных Технологий - дистанционное образование [Электронный ресурс]. - Режим доступа : <http://www.intuit.ru/>.
24. CodeNet - все для программиста [Электронный ресурс]. - Режим доступа: <http://www.codenet.ru/>.

25. Computer Science [Электронный ресурс]. - Режим доступа : <http://www.lektorium.tv>.
26. MSDN-WindowsAPI [Электронный ресурс]. - Режим доступа : [http://www.vsokovikov.narod.ru/New MSDN API/index\\_msdn.htm](http://www.vsokovikov.narod.ru/New_MSDN_API/index_msdn.htm).
27. REALCODING - для настоящих программистов [Электронный ресурс]. - Режим доступа : <http://www.realcoding.net/>.
28. RSDN [Электронный ресурс]. - Режим доступа : <http://www.rsdn.ru/>.