

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ СЕМЕНА КУЗНЕЦЯ**

ПРОГРАМУВАННЯ

**Робоча програма
для студентів спеціальності
122 "Комп'ютерні науки та інформаційні технології"
першого (бакалаврського) рівня**

**Харків
ХНЕУ ім. С. Кузнеця
2017**

УДК 004.42(07.034)

П 79

Затверджено на засіданні кафедри інформаційних систем.
Протокол № 16 від 30.06.2016 р.

Самостійне електронне текстове мережеве видання

Укладачі: В. М. Федорченко
О. В. Щербаков
Ю. Е. Парфьонов

Програмування : робоча програма для студентів спеціальності 122 "Комп'ютерні науки та інформаційні технології" першого (бакалаврського) рівня : [Електронне видання] / уклад. В. М. Федорченко, О. В. Щербаков, Ю. Е. Парфьонов. – Харків : ХНЕУ ім. С. Кузнеця, 2017. – 56 с.

Подано тематичний план навчальної дисципліни та її зміст за модулями й темами. Вміщено плани лекцій та лабораторних занять, матеріали для закріплення знань (завдання для самостійної роботи, контрольні запитання), методичні рекомендації щодо оцінювання знань студентів, професійні компетентності, якими повинен володіти студент після вивчення дисципліни.

Рекомендовано для студентів спеціальності 122 "Комп'ютерні науки та інформаційні технології" першого (бакалаврського) рівня всіх форм навчання.

УДК 004.42(07.034)

© Харківський національний економічний
університет імені Семена Кузнеця, 2017

Вступ

Дисципліну "Програмування" віднесено до групи освітньо-професійних дисциплін підготовки бакалаврів за спеціальністю 122 "Комп'ютерні науки та інформаційні технології".

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 122 "Комп'ютерні науки та інформаційні технології" усіх форм навчання.

1. Опис навчальної дисципліни

Найменування показників	Галузь знань, напрям підготовки, освітній ступінь	Характеристика навчальної дисципліни	
		денна форма навчання	заочна форма навчання
Кількість кредитів – 5 (7 – заочна)	Галузь знань 12 "Інформаційні технології"	Базова	
Змістових модулів – 2	Спеціальність: 122 "Комп'ютерні науки та інформаційні технології"	Рік підготовки	
Загальна кількість годин – 150 (210)		1-й	1-й
		Семестр	
		1-й	1-й, 2-й
Тижневих годин для денної форми навчання: аудиторних – 4; самостійної роботи студента – 6	Освітній ступінь: бакалавр	32 год	16 год
		Лабораторні	
		32 год	20 год
		Самостійна робота	
		83 год	171 год
		Вид контролю	
		екзамен	
3 год	3 год		

Примітка. Співвідношення кількості годин аудиторних занять до самостійної й індивідуальної роботи становить:

- для денної форми навчання – 74 %;
- для заочної форми навчання – 21 %.

2. Мета та завдання навчальної дисципліни

Програма вивчення нормативної навчальної дисципліни "Програмування" складена відповідно до освітньо-професійної програми підготовки бакалавра за спеціальністю 122 "Комп'ютерні науки та інформаційні технології".

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.

Об'єктом навчальної дисципліни є програмне забезпечення інформаційних систем.

Предметом вивчення навчальної дисципліни є теорія і практика застосування у програмуванні базових алгоритмічних структур і базових структур даних на базі сучасних технологій розроблення програмного забезпечення.

Міждисциплінарні зв'язки. Навчальна дисципліна "Програмування" є інструментальною основою для виконання аналітичної частини подальших спецкурсів, а також курсових і дипломних робіт. Для успішного вивчення дисципліни необхідні базові знання, отримані студентами в обсязі шкільної програми, а також поточні знання під час паралельного освоєння ними дисциплін "Вступ до комп'ютерних наук", "Вища математика", "Дискретна математика". Навчальна дисципліна "Програмування" 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", "Кросплатформене програмування", "Програмування для мобільних пристроїв", "Технології захисту інформації", "Дипломне проектування".

Основними **завданнями** вивчення навчальної дисципліни є:

вивчення основних етапів процесу проектування програмного забезпечення і визначення принципів процедурного програмування щодо розроблення програм мовою C/C++;

вивчення типових підходів до розроблення й аналізу найбільш розповсюджених алгоритмів вирішення економіко-математичних задач;

здійснення аналізу можливостей сучасних інструментальних середовищ розроблення програм (на прикладі середовища Visual C++.NET); визначення концепцій і вивчення основних принципів організації програм.

У результаті вивчення навчальної дисципліни студент повинен:

знати:

основні етапи процесу проектування програмного забезпечення; абетку, синтаксис і семантику мов програмування C/C++;

типові алгоритмічні конструкції і відповідні оператори мов програмування C/C++;

принципи процедурного і структурного програмування;

особливості застосування сучасних базових інструментальних програмних засобів, призначених для вирішення економічних задач;

базові та похідні типи даних;

оператори управління програмою;

команди передпроцесорної обробки;

правила роботи з функціями;

правила роботи з масивами;

початкові основи системи вводу-виводу C/C++;

основні принципи роботи з DLL;

вміти:

вирішувати задачі з курсу вищої математики (чисельне диференціювання та інтеграція, вирішення рівнянь і т. д.);

створювати і обробляти одновимірні та багатовимірні масиви;

проводити передпроцесорну обробку файлів програми, стандартних функцій та функцій користувача;

проводити найпростіший ввід-вивід даних;

створювати і використовувати DLL;

використовувати сучасне інструментальне програмне забезпечення;

користуватися раніше складеними програмами і здійснювати супровід програм;

вносити зміни в програму;

виконувати владження програм за допомогою вбудованих інструментальних засобів.

Формування суджень

Після вивчення навчальної дисципліни, студент має демонструвати здатність критично вивчати, аналізувати та оцінювати з різних точок зору:

технології, процеси, методи та інструменти для розроблення програмних продуктів;

особисті знання і потребу в інших знаннях;

перспективи та загальні тенденції щодо розроблення програмних продуктів.

Комунікація

Після успішного закінчення курсу, студент повинен бути в змозі продемонструвати здатність планувати і проводити гарні презентації, а також описувати прийняті рішення з використанням усного, символічного та письмового мовлення.

Здатність до подальшого навчання, розвитку

Після успішного закінчення курсу, студент повинен бути в змозі продемонструвати здатність вивчати нові технології, методи та прийоми щодо розроблення ПЗ, а також критично аналізувати їх для професійної роботи і подальшого використання.

Програма навчальної дисципліни складається з двох змістових модулів:

змістовий модуль 1. Організація програм;

змістовий модуль 2. Основні похідні типи даних мов C/C++.

У процесі викладання навчальної дисципліни основна увага приділяється оволодінню студентами професійними **компетентностями** та їх складовими, що наведено в табл. 2.1.

Таблиця 2.1

Професійні компетентності, які отримують студенти після вивчення навчальної дисципліни

Код компетентності	Назва компетентності	Складові компетентності
1	2	3
П* 1	Приймати оптимальні рішення щодо розроблення ПЗ	Визначати цільові програмні платформи
		Визначати інструментальні засоби розроблення ПЗ
		Визначати оптимальні методи розроблення ПЗ

1	2	3
П 2	Розробляти ефективні алгоритми та програми з використанням економіко-математичних методів та моделей	Визначати способи формалізації та реалізації ПЗ
		Здійснювати обґрунтований вибір економіко-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 ПЗ
		Здійснювати оцінювання отриманих результатів

* Програмування

Структуру складових професійних компетентностей та їх формування відповідно до Національної рамки кваліфікацій України наведено в табл. А.1 додатка А.

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 Visual C++.NET. Платформа DOT.NET. 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/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 (sizeof). 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 do-while, оператор for). Вкладені цикли. Керівні оператори в циклах: оператор break, оператор continue, оператор goto. Рекомендації щодо вибору циклів.

Тема 6. Передпроцесорна 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 typedef. Оператор 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/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 new і delete. Динамічні масиви. Динамічні масиви як параметри функцій.

Тема 12. Введення в систему вводу-виводу C/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.

4. Структура навчальної дисципліни

Із самого початку вивчення навчальної дисципліни студент має бути ознайомлений як з робочою програмою навчальної дисципліни і формами організації навчання, так і зі структурою, змістом та обсягом кожного з її змістових модулів, а також з усіма видами контролю та методикою оцінювання сформованих професійних компетентностей.

Вивчення студентом навчальної дисципліни відбувається шляхом послідовного і ґрунтовного опрацювання змістових модулів. Змістовий модуль – це окремий, відносно самостійний блок дисципліни, який логічно об'єднує кілька навчальних елементів дисципліни за змістом та взаємозв'язками. Тематичний план дисципліни складається з двох змістових модулів (табл. 4.1).

Таблиця 4.1

Структура залікового кредиту навчальної дисципліни

Назви змістових модулів і тем	Кількість годин											
	денна форма						заочна форма					
	усього	у тому числі					усього	у тому числі				
		лекції	екзамен	лабораторні	індивідуальні	самостійна робота		лекції	екзамен	лабораторні	індивідуальні	самостійна робота
1	2	3	4	5	6	7	8	9	10	11	12	13
Змістовий модуль 1. Організація програм												
<i>Тема 1.</i> Етапи розроблення та впровадження програм. Поняття алгоритму та типові алгоритмічні структури програмування	20	2	–	2	10	6	28	1	–	2	20	5
<i>Тема 2.</i> Архітектура комп'ютерів, принципи Джона фон Неймана	2	–	–	–	–	2	4	–	–	–	–	4
<i>Тема 3.</i> Позиційні системи числення	3	–	–	–	–	3	5	–	–	–	–	5
<i>Тема 4.</i> Елементи алгоритмічних мов: концепція типів даних, імена, значення, покажчики, змінні, константи, операції, вирази	12	2	–	4	–	6	14	2	–	2	–	10

1	2	3	4	5	6	7	8	9	10	11	12	13
Тема 5. Структурне програмування: послідовність, розгалуження та цикли	16	2	–	8	–	6	19	3	–	4	–	12
Тема 6. Передпроцесорна обробка	8	2	–	2	–	4	8	1	–	2	–	5
Тема 7. Процедурно-орієнтоване програмування. Рекурсія	12	4	–	2	–	6	19	2	–	2	–	15
Тема 8. Бібліотеки динамічного компонування (DLL)	9	1	–	2	–	6	11	1	–	–	–	10
Тема 9. Методології розроблення програм: низхідне та висхідне проектування, модульне програмування	3	1	–	–	–	2	5	–	–	–	–	5
Контрольна робота за темами 1 – 9	5	2	–	–	–	3	–	–	–	–	–	–
Разом за змістовим модулем 1	90	16	–	20	10	44	113	10	–	12	20	71
Змістовий модуль 2. Основні похідні типи даних мов програмування C/C++												
Тема 10. Масиви	20	6	–	8	–	6	28	2	–	6	–	20
Тема 11. Похідні типи даних. Рядки в стилі C	16	6	–	4	–	6	24	2	–	2	–	20
Тема 12. Введення в систему вводу-виводу C/C++	7	2	–	–	–	5	22	2	–	–	–	20
Контрольна робота за темами 10 – 12	4	2	–	–	–	2	–	–	–	–	–	–
Разом за змістовим модулем 2	47	16	–	12	–	19	74	6	–	8	–	60
Підготовка до екзамену	8	–	–	–	–	8	18	–	–	–	–	18
Передекзаменаційні консультації	2	–	–	–	–	2	2	–	–	–	–	2
Екзамен	3	–	3	–	–	–	3	–	3	–	–	–
Усього годин	150	32	3	32	10	73	210	16	3	20	20	151

5. Теми лабораторних занять

Лабораторне заняття – це форма навчального заняття, спрямована на закріплення та вдосконалення студентом теоретичних знань, отриманих як на лекційних, практичних заняттях, так і в процесі самостійного вивчення матеріалу. Під час лабораторного заняття студенти під керівництвом викладача особисто набувають практичних навичок у роботі

з обчислювальною технікою, оволодівають методикою створення програмних продуктів у програмному середовищі.

На лабораторному занятті студенти під керівництвом викладача проводять досліді в навчальних лабораторіях з використанням комп'ютерної техніки.

Основною метою лабораторного заняття є практичне підтвердження окремих теоретичних положень та набуття практичних умінь з виконання обчислювальних експериментів. Головна особливість цих занять полягає в тому, що вони об'єднують теорію з практикою, забезпечують їх єдність. Сукупність лабораторних занять із дисципліни є лабораторним практикумом, що спланований за єдиним задумом. Лабораторні заняття плануються після проведення лекцій. А за необхідності розроблення програм, проектування баз даних або підготовки складних розрахунків і початкових даних, перед лабораторними заняттями проводяться практичні.

Напередодні проведення кожного лабораторного заняття (звичайно після відповідної лекції, практичного (індивідуального) заняття) студентам видається завдання, що містить:

- тему і мету заняття;

- скорочені теоретичні відомості щодо змісту лабораторного заняття;

- список питань для підготовки (це можуть бути контрольні питання з теми, що вивчається, заповнення роздаткових матеріалів індивідуальними даними, розроблення програм, таблиць і т. д.);

- послідовність належних виконанню дій (завдання на лабораторну роботу);

- вимоги до змісту звіту.

Студент повинен вивчити навчальний матеріал, завдання, підготувати необхідні для роботи на занятті матеріали і знати відповіді на контрольні питання. У ході підготовки може бути створена заготівка звіту, що дозволить заощадити час на занятті.

Лабораторні заняття проводяться в аудиторіях ОЦ шляхом розроблення програм (проектів) розв'язання індивідуальних завдань (задач).

Усі лабораторні заняття з дисципліни проводяться фронтально, кожний студент працює за окремим комп'ютером.

На початку заняття, після оголошення теми, цільової установки, коротких вказівок щодо особливостей роботи викладачем проводиться контроль підготовленості студентів, звичайно, шляхом перевірки відповідей на контрольні питання, рідше, у формі усної бесіди з теми заняття.

Для контролю може використовуватися і тестування. Обов'язково перевіряється наявність матеріалів для виконання роботи (програм, роздаткового матеріалу з відпрацьованими індивідуальними питаннями, початкових даних для вирішення задач, заготівок звіту і т. д.).

За відсутності матеріалів, необхідних для виконання роботи, і знань, які не дозволяють виконати роботу, студент до роботи не допускається, і йому пропонується виконати необхідну підготовку. Сама робота повинна виконуватися в додатковий час.

У ході заняття студенти самостійно виконують передбачені завдання дії, заносючи результати у звіт. На це відводиться до 85 – 90 % часу заняття. Викладач здійснює контроль за роботою і надає допомогу під час виникнення утруднень, звертає увагу на складні, ключові моменти. Причому основну увагу надають не вказівкою на конкретну помилку, а методикою пошуку причин виникнення цих помилок.

Складання звіту – це відповідальний етап лабораторного заняття. Під час його складання студенти розвивають навички аналізу, узагальнення і творчого осмислення результатів роботи, а також навички розробки документації до програмного продукту.

Необхідно прагнути до того, щоб студенти оформляли звіт про виконану роботу і подавали його викладачу до кінця лабораторної роботи. Цьому сприяє наявність наперед підготовленої заготівки, в яку послідовно заносяться всі необхідні дані та зроблені висновки. Звіт повинен бути поданий у вигляді електронного документа.

За наслідками контролю готовності до роботи, обсягу і правильності її виконання, повноти та якості оформлення звіту і його захисту, терміну захисту викладач виставляє оцінку. Звіти, які не подані під час заняття, захищаються в додатковий час. В окремих випадках оцінка може виставлятися за групу взаємозв'язаних робіт.

Під час оцінювання лабораторної роботи викладач враховує правильність та розуміння роботи розроблених програмних продуктів, уміння працювати у програмному середовищі. Оцінки за кожну лабораторну роботу вносяться у відповідний журнал.

Студент, що пропустив лабораторне заняття або не допущений до нього, зобов'язаний виконати відповідну роботу під час самостійної підготовки і відзвітувати. Повторне здавання робіт, які не були прийняті, проводиться під час додаткових занять або під час наступних лабораторних занять.

Оцінки, отримані за окремі лабораторні заняття враховуються під час виставлення поточної модульної оцінки з навчальної дисципліни.

У процесі лабораторного заняття викладач організує такі види методичної роботи зі студентами:

- вирішення поточних індивідуальних завдань на лабораторну роботу;
- перевірку завдань щодо розроблення програм та алгоритмів;
- захист лабораторних робіт окремих студентів.

Перелік тем лабораторних занять наведено в табл. 5.1.

Таблиця 5.1

Перелік тем лабораторних занять

Назва теми	Теми лабораторних занять	Кількість годин	Література
1	2	3	4
Змістовий модуль 1. Організація програм			
<i>Тема 1.</i> Етапи розроблення та впровадження програм. Поняття алгоритму та типові алгоритмічні структури програмування	<i>Лабораторна робота 1.</i> Розроблення графічних схем алгоритмів методом покрокової деталізації	2	Основна: [3; 5; 7; 8; 10; 11]. Додаткова: [18, 20, 24]
<i>Тема 4.</i> Елементи алгоритмічних мов: концепція типів даних, імена, значення, покажчики, змінні, константи, операції, вирази	<i>Лабораторна робота 2.</i> Знайомство з інтегрованим середовищем розроблення програм MS Visual C++.NET. Підготовка і розв'язання на ПК задач лінійного характеру	4	Основна: [3; 5; 7; 8; 10; 11]. Додаткова: [18; 20; 24]
<i>Тема 5.</i> Структурне програмування: послідовність, розгалуження та цикли	<i>Лабораторна робота 3.</i> Підготовка і розв'язання на ПК задач з розгалуженням.	4	Основна: [3; 5; 7; 8; 10; 11]. Додаткова: [18; 20; 22; 24]
	<i>Лабораторна робота 4.</i> Підготовка і розв'язання на ПК задач з використанням циклів	4	
<i>Тема 6.</i> Передпроцесорна обробка	<i>Лабораторна робота 5.</i> Підготовка і розв'язання на ПК задач з використанням функцій і макросів	6	Основна: [4; 7; 8; 10; 11]. Додаткова: [18; 20; 22; 24]

1	2	3	4
Тема 7. Процедурно-орієнтоване програмування. Рекурсія			Основна: [3; 5; 7; 8; 10; 11]. Додаткова: [18; 20; 22; 24]
Тема 8. Бібліотеки динамічного компонування(DLL)			Основна: [4; 5; 7; 8; 10; 11]. Додаткова: [18; 20; 22; 24]
Змістовий модуль 2. Основні похідні типи даних мов програмування C/C++			
Тема 10. Масиви	Лабораторна робота 6. Підготовка і розв'язання на ПК задач обробки одновимірних масивів.	4	Основна: [3; 5; 7; 8; 10; 11]. Додаткова: [18; 20; 22; 24]
	Лабораторна робота 7. Підготовка і розв'язання на ПК задач обробки двовимірних масивів	4	
Тема 11. Похідні типи даних. Рядки в стилі C	Лабораторна робота 8. Підготовка і розв'язання на ПК задач обробки масивів з використанням покажчиків	4	Основна: [3; 5; 7; 8; 10; 11]. Додаткова: [18; 20; 22; 24]

Приклади типових лабораторних завдань

Змістовий модуль 1

Скласти програму для знаходження значень виразів А, В по заданих значеннях початкових даних x, y, z.

$$A = 2^{-x} \sqrt{x + 4\sqrt{|y|}}, B = \sqrt[3]{e^{x-1/\sin z}}$$

Змістовий модуль 2

Скласти програму, яка виконує обробку масива у функції та реалізує такі дії.

У одновимірному масиві, що складається з n дійсних елементів, обчислити:

- 1) суму негативних елементів масиву;
- 2) добуток елементів масиву, розташованих між максимальним і мінімальним елементами.

Упорядкувати елементи масиву за збільшенням.

6. Самостійна робота

Самостійна робота студента (СРС) – це форма організації навчального процесу, за якої заплановані завдання виконуються студентом самостійно під методичним керівництвом викладача.

Мета СРС – засвоєння в повному обсязі навчальної програми та формування у студентів загальних і професійних компетентностей, які відіграють суттєву роль у становленні майбутнього фахівця вищого рівня кваліфікації.

Навчальний час, відведений для самостійної роботи студентів денної форми навчання, визначається навчальним планом і становить 57 % (86 годин) від загального обсягу навчального часу на вивчення дисципліни (150 годин). У ході самостійної роботи студент має перетворитися на активного учасника навчального процесу, навчитися свідомо ставитися до оволодіння теоретичними і практичними знаннями, вільно орієнтуватися в інформаційному просторі, нести індивідуальну відповідальність за якість власної професійної підготовки. СРС включає: опрацювання лекційного матеріалу; опрацювання та вивчення рекомендованої літератури, основних термінів та понять за темами дисципліни; підготовку до лабораторних занять; поглиблене опрацювання окремих лекційних тем або питань; виконання індивідуальних завдань (вирішення розрахункових індивідуальних та комплексних завдань); пошук (підбір) та огляд літературних (електронних) джерел за заданою проблематикою дисципліни; аналітичний розгляд наукової публікації; контрольну перевірку студентами особистих знань за запитаннями для самодіагностики; підготовку до контрольних робіт та інших форм поточного контролю; систематизацію вивченого матеріалу з метою підготовки до семестрового екзамену.

Необхідним елементом успішного засвоєння матеріалу навчальної дисципліни є самостійна робота студентів з вітчизняною та закордонною спеціальною літературою, електронними джерелами. Основні види самостійної роботи, які запропоновані студентам для засвоєння теоретичних знань з навчальної дисципліни, наведено в табл. 6.1.

Завдання для самостійної роботи студентів та форми її контролю

Назва теми	Зміст самостійної роботи студентів	Кількість годин	Форми контролю СРС	Література
1	2	3	4	5
Змістовий модуль 1. Організація програм				
<i>Тема 1.</i> Етапи розроблення та впровадження програм. Поняття алгоритму та типові алгоритмічні структури програмування	Вивчення лекційного матеріалу, підготовка до лабораторного заняття. Виконання індивідуального завдання "Алгоритмізація циклічних обчислювальних процесів"	16	Захист лабораторної роботи. Захист індивідуального завдання	Основна: [1; 2]. Додаткова: [18; 20; 24]
<i>Тема 2.</i> Архітектура комп'ютерів, принципи Джона фон Неймана	Вивчення лекційного матеріалу. Пошук, підбір та огляд літературних (електронних) джерел за заданою тематикою	2	Презентація	Основна: [3; 5; 7; 10; 11]. Додаткова: [18; 20; 24]
<i>Тема 3.</i> Позиційні системи числення	Вивчення лекційного матеріалу. Пошук, підбір та огляд літературних (електронних) джерел за заданою тематикою	3	Презентація	Основна: [3; 5; 7; 8; 10; 11]. Додаткова: [18; 20; 24]
<i>Тема 4.</i> Елементи алгоритмічних мов: концепція типів даних, імена, значення, покажчики, змінні, константи, операції, вирази	Вивчення лекційного матеріалу, підготовка до лабораторного заняття	6	Захист лабораторної роботи	Основна: [1 – 3]. Додаткова: [4; 8; 9; 11; 12; 16; 17; 22; 25]
<i>Тема 5.</i> Структурне програмування: послідовність, розгалуження та цикли	Вивчення лекційного матеріалу, підготовка до лабораторного заняття	6	Захист лабораторної роботи	Основна: [3; 5; 7; 8; 10; 11]. Додаткова: [18; 20; 22; 24]
<i>Тема 6.</i> Передпроцесорна обробка	Вивчення лекційного матеріалу, підготовка до лабораторного заняття	4	Захист лабораторної роботи	Основна: [4; 7; 10; 11]. Додаткова: [18; 20; 22; 24]

Продовження табл. 6.1

1	2	3	4	5
<i>Тема 7.</i> Процедурно-орієнтоване програмування. Рекурсія	Вивчення лекційного матеріалу, підготовка до лабораторного заняття	6	Захист лабораторної роботи	Основна: [3; 5; 7; 10; 11]. Додаткова: [18; 20; 22; 24]
<i>Тема 8.</i> Бібліотеки динамічного компонування (DLL)	Вивчення лекційного матеріалу, підготовка до лабораторного заняття	6	Захист лабораторної роботи	Основна: [4; 5; 7; 10; 11]. Додаткова: [18; 20; 22; 24]
<i>Тема 9.</i> Методології розроблення програм: низхідне та висхідне проектування, модульне програмування	Вивчення лекційного матеріалу. Пошук, підбір та огляд літературних (електронних) джерел за заданою тематикою	2	Захист лабораторної роботи	Основна: [5; 9]. Додаткова: [22; 24]
Контрольна робота за темами 1 – 9	Вивчення лекційного матеріалу. Пошук, підбір та огляд літературних (електронних) джерел за заданою тематикою. Підготовка до контрольної роботи	3	Результати контрольної роботи	Основна: [3; 5; 7; 10; 11]. Додаткова: [18; 20; 22; 24]
Усього за змістовим модулем 1		54		
Змістовий модуль 2. Основні похідні типи даних мов програмування C/C++				
<i>Тема 10.</i> Масиви	Пошук, підбір та огляд літературних джерел за заданою тематикою. Вивчення лекційного матеріалу, підготовка до лабораторного заняття	6	Захист лабораторної роботи	Основна: [3 – 7]. Додаткова: [16; 18 – 25]
<i>Тема 11.</i> Похідні типи даних. Рядки в стилі C	Пошук, підбір та огляд літературних джерел за заданою тематикою. Вивчення лекційного матеріалу, підготовка до лабораторного заняття	6	Захист лабораторної роботи	Основна: [3 – 7; 11]. Додаткова: [13 – 16; 21 – 25]

1	2	3	4	5
Тема 12. Введення в систему вводу-виводу C/C++	Пошук, підбір та огляд літературних джерел за заданою тематикою. Вивчення лекційного матеріалу, підготовка до лабораторного заняття	5	Експрес-опитування	Основна: [3 – 7; 11]. Додаткова: [13 – 16; 21 – 25]
Контрольна робота за темами 10 – 12	Вивчення лекційного матеріалу. Пошук, підбір та огляд літературних (електронних) джерел за заданою тематикою. Підготовка до контрольної роботи	2	Результати контрольної роботи	Основна: [3 – 7; 11]. Додаткова: [13 – 16; 21 – 25]
Усього за змістовим модулем 2		19		
Підготовка до екзамену (Передекзаменаційні консультації)		10		Основна: [1 – 3; 5 – 7; 10]. Додаткова: [18 – 28]
Екзамен		3		
Усього за модулем		86		

6.1. Індивідуальне завдання

Індивідуальне завдання (ІЗ) студента є вибіркоким видом позааудиторної самостійної роботи студента та має навчально-дослідницький характер, виконується у процесі вивчення програмного матеріалу навчальної дисципліни і завершується разом зі складанням підсумкового екзамену з цієї навчальної дисципліни. Виконання ІЗ є одним із важливих засобів підвищення якості підготовки майбутніх спеціалістів, які здатні застосовувати на практиці теоретичні знання, вміння та навички з цієї навчальної дисципліни.

Підготовка ІЗ передбачає систематизацію, закріплення, розширення теоретичних і практичних знань із дисципліни та застосування їх у процесі розв'язання конкретних економічних ситуацій, розвиток навичок

самостійної роботи й оволодіння методикою дослідження та експерименту, пов'язаних із темою ІЗ.

ІЗ передбачає наявність таких елементів наукового дослідження: практичної значущості, комплексного системного підходу до вирішення завдань дослідження, теоретичного використання передової сучасної методології та наукових розробок, наявність елементів творчості, вміння застосовувати сучасні технології.

Практична значущість ІЗ полягає в обґрунтуванні реальності її результатів для потреб практики.

Реальною вважається робота, яка виконана відповідно до наявних проблем підприємства, на основі його реальних даних за кілька років і результати якої повністю або частково можуть бути впроваджені в практику діяльності підприємства.

Комплексний системний підхід до розкриття теми роботи полягає в тому, що предмет дослідження розглядається з різних точок зору – з позицій теоретичної бази і практичних напрацювань, умов його реалізації на підприємстві, аналізу, обґрунтування шляхів удосконалення тощо – в тісному взаємозв'язку та єдиній логіці викладу.

Застосування сучасної методології полягає в тому, що в процесі виконання аналізу стратегічних позицій підприємства й обґрунтування шляхів удосконалення окремих аспектів предмета та об'єкта дослідження студент повинен використовувати відомості про новітні досягнення в техніці і технологіях дослідження, застосовувати різноманітні методи й засоби діагностичних досліджень, підходи до визначення та обґрунтування вибору критеріїв і показників експрес-діагностування виробничо-економічної системи або її елементів.

У процесі виконання ІЗ, разом з теоретичними знаннями і практичними навичками за фахом, студент повинен продемонструвати здатність до науково-дослідної роботи та вміння творчо мислити.

Тема: "Алгоритмізація циклічних обчислювальних процесів".

Мета роботи: набуття практичних навичок розроблення алгоритмів.

Основні завдання:

1) розроблення графічної схеми алгоритму розв'язання практичної задачі.

Тема: "Обробка матриць".

Мета роботи: набуття практичних навичок розроблення алгоритмів і програм обробки масивів даних.

Основні завдання:

- 1) розроблення графічної схеми алгоритму розв'язання практичної задачі;
- 2) розроблення та відлагодження програми розв'язання практичної задачі.

Індивідуальне завдання виконується студентом самостійно за консультування з викладачем протягом вивчення дисципліни відповідно до графіка навчального процесу.

Індивідуальне завдання видається викладачем на початку семестру, протягом якого вивчається дисципліна. Студент має надати ІЗ для перевірки наприкінці семестру, але не пізніше терміну проведення підсумкового модульного контролю. Бали за виконання ІЗ враховуються під час виставлення загальної оцінки з дисципліни.

Вимоги до оформлення. Формат аркуша – А4. Поля: ліве – 30 мм; праве, верхнє та нижнє – 20 мм. Шрифт – Arial, розмір шрифту для основного тексту – кг. 14. Міжрядковий інтервал – множник 1,3. Не дозволяються виділення в тексті курсивом та підкреслення.

Обсяг ІЗ повинен становити у друкованому варіанті 3 – 4 сторінки.

Кожен розділ розпочинають з нової сторінки. Назви розділів оформляються великими літерами по центру сторінки. Сторінки нумеруються у правому верхньому куті. На титульному аркуші номер сторінки не ставиться.

Вимоги до змісту. Індивідуальне науково-дослідне завдання повинне містити такі розділи.

Титульна сторінка. Повинна містити назву університету; назву кафедри; назву навчальної дисципліни; тему ІЗ із вказівкою бази дослідження; прізвище, ініціали студента, назву навчальної дисципліни, номер академічної групи; дату подання ІЗ викладачеві на перевірку (день, місяць, рік).

Зміст. Повинен відтворювати назви розділів, параграфів тощо, які розкривають тему ІЗ, із зазначенням номерів сторінок, на яких вони розміщені.

Вступ. У "Вступі" студентом розкривається сутність і стан наукової задачі та її значущість, засоби її вирішення, вхідні та вихідні дані для розроблення теми ІЗ.

Основна частина. Складається з 3-х розділів.

Перший розділ повинен містити графічну схему алгоритма розв'язання задачі та опис інтерфейсу користувача, в цьому розділі студент повинен визначити:

- 1) основні та допоміжні (якщо необхідно) схеми алгоритмів;
- 2) сценарій взаємодії користувача та програми.

Другий розділ повинен містити код програми на мові C++ з відповідними коментарями.

Третій розділ повинен містити результати тестування програми, результати роботи програми з реальними вихідними даними.

Висновки. Викладають аналіз отриманих результатів. Далі формулюють пропозиції щодо практичного використання результатів ІЗ.

Список використаної літератури слід розмішувати в алфавітному порядку прізвищ перших авторів або заголовків. Бібліографічний опис джерел складають відповідно до ДСТУ ГОСТ 7.1:2006 "Бібліографічний запис, бібліографічний опис. Загальні вимоги та правила складання".

Додатки можуть бути включені за необхідності. У разі наявності кількох додатків оформляється окрема сторінка "ДОДАТКИ", номер якої є останнім, що включається до обсягу ІЗ. Кожен додаток починають із нової сторінки. Відповідно до вимог, додаток називають у такий спосіб: "Додаток А", "Додаток Б" і т. д. за алфавітом, за винятком букв Г, Є, І, Ї, Й, О, Ч, Ь, при цьому написи "Додаток...", "Продовження додатка..." та "Закінчення додатка..." пишуться малими літерами з першої великої і вирівнюються по правому краю сторінки. Допускається розподіл додатків на розділи типу "Додаток А.3". Ілюстрації, таблиці, формули нумерують відповідно до назви додатка, наприклад "рис. А.3".

6.2. Контрольні запитання для самоперевірки

1. Структура типової програми на мові C++.
2. Бібліотечні файли, їх організація і використання.
3. Опишіть можливості застосування інтегрованого середовища Visual C++ .NET для розробки застосувань.
4. Дайте класифікацію типів даних, способи опису їх у програмах, приклади використання.
5. Перерахуйте структури мови C++, що керують, і вкажіть особливості їх застосування.
6. Формат і робота умовного оператора в повній і скороченій формах. Наведіть приклади.

7. Формат і робота оператора **switch**. Наведіть приклади.
8. Формат і робота оператора циклу **while** (цикл з передумовою). Наведіть приклади.
9. Формат і робота оператора циклу **do/while** (цикл з післяумовою). Наведіть приклади.
10. Формат і робота оператора циклу **for**. Наведіть приклади.
11. Як записати два і більше виразів в умові **while**?
12. Чи відрізняється умова циклу в циклі **while** від умови циклу **do/while**?
13. Дайте визначення поняття "масив". До яких типів даних відносяться масиви в C++?
14. Наведіть приклад циклу **for** від 0 до 20. Коли виконується кожний з трьох виразів?
15. Де в заголовку **for** може використовуватися кома?
16. Чи можна опускати крапки з комою в циклі **for**?
17. Як працює оператор **break**?
18. Як працює оператор **continue**?
19. Як нумеруються елементи масиву `a[5]`?
20. Чи може розмірність масиву визначатися змінній?
21. Як ініціалізувати масив? Чи можна при цьому опускати розмірність?
22. Що буде, якщо в списку ініціалізації менше елементів, ніж розмірність масиву? А якщо більше?
23. Чи можна ініціалізувати символічний масив рядком символів?
24. Чи можна привласнити масив масиву?
25. Визначте поняття покажчика в C++. Чим відрізняються покажчики від посилань?
26. Що відбудеться, якщо буде вихід за межу масиву?
27. Що значить запис `a[1]` для масиву `a[5][5]`?
28. Що тут визначене? `int *i1, i2.`
29. Як визначається операція взяття адреси в C++?
30. Що таке покажчик типу `void*` ? Чи можна його розіменувати?
31. Що позначає окремо ім'я `a` для масиву `a[10]`?
32. На що вказує `ptr + 10`, якщо `ptr` вказує на перший елемент масиву?
33. Чим відрізняється робота з масивом від роботи з покажчиком?
34. Як розмістити в C++ об'єкт у вільній пам'яті?

35. Як звільнити пам'ять об'єкта?

36. Опишіть структури даних – масиви. Поясніть організацію багатомірного масиву. Для чого використовуються покажчики і посилання?

37. Програми модулі C++ (функції). Визначення функцій, виклик, прототипи.

38. Створіть функцію, яка приймає три аргументи: ім'я масиву елементів типу int, розмір масиву і значення типу int. Функція повинна встановити для кожного елементу масиву значення типу int.

39. Назвіть засоби форматування вводу-виводу.

40. Перерахуйте основні директиви компілятора.

7. Індивідуально-консультативна робота

Індивідуально-консультативна робота здійснюється за графіком індивідуально-консультативної роботи у формі індивідуальних занять, консультацій, перевірки виконання індивідуальних завдань, перевірки та захисту завдань, що винесені на поточний контроль, тощо.

Формами організації індивідуально-консультативної роботи є:

а) за засвоєнням теоретичного матеріалу:

консультації: індивідуальні (запитання – відповідь), групові (розгляд типових прикладів – ситуацій);

б) за засвоєнням практичного матеріалу:

консультації індивідуальні та групові;

в) для комплексної оцінки засвоєння програмного матеріалу:

індивідуальне здавання виконаних робіт.

8. Методи навчання

Під час викладання дисципліни передбачено застосування активних та інтерактивних методів навчання – ділових ігор, рольових ігор, тренінгів, семінарів в активній формі, розгляд кейсів, модерації. До активних, інтерактивних методів навчання також відносяться методи дистанційного навчання (електронні підручники, навчальні системи). Основні відмінності активних та інтерактивних методів навчання від традиційних визна-

чаються не тільки методикою і технікою викладання, але і високою ефективністю навчального процесу, який виявляється у:

- високій мотивації студентів;
- закріпленні теоретичних знань на практиці;
- підвищенні самосвідомості студентів;
- формуванні здатності ухвалювати самостійні рішення;
- формуванні здатності до колективних рішень;
- формуванні здатності до соціальної інтеграції;
- придбанні навичок вирішення конфліктів;
- розвитку здатності до компромісів.

Розподіл форм та методів активізації процесу навчання за темами навчальної дисципліни наведено в табл. 8.1 і 8.2.

Таблиця 8.1

**Розподіл форм та методів активізації процесу навчання
за темами навчальної дисципліни**

Тема	Практичне застосування навчальних технологій
1	2
<i>Тема 1.</i> Етапи розроблення та впровадження програм. Поняття алгоритму та типові алгоритмічні структури програмування	Семінар-дискусія, мозкова атака з питання "Розроблення графічних схем алгоритмів". Презентація результатів роботи в малих групах
<i>Тема 4.</i> Елементи алгоритмічних мов: концепція типів даних, імена, значення, покажчики, змінні, константи, операції, вирази	Міні-лекція з питання "Способи завдання логічних функцій у мові C++". Модерація з питання "Подання даних у пам'яті ПК"
<i>Тема 5.</i> Структурне програмування: послідовність, розгалуження та цикли	Проблемна лекція, з питання "Організація циклічних процесів". Кейс-метод з питання "Розширення можливостей циклічних програм"
<i>Тема 7.</i> Процедурно-орієнтоване програмування. Рекурсія	Проблемна лекція з питання "Використання процедурного програмування". Презентація результатів роботи в малих групах
<i>Тема 10.</i> Масиви	Проблемна лекція з питання "Порівняння алгоритмів сортування масивів".
<i>Тема 11.</i> Похідні типи даних. Рядки в стилі C	Міні-лекція, семінар-дискусія з питання "Оптимальна обробка символічної інформації"
<i>Тема 12.</i> Введення в систему вводу-виводу C/C++	Проблемна лекція з питання "Шляхи організації інтерфейсу користувача". Модерація "Обробка помилок вводу даних"

Лекції проблемного характеру – один із найважливіших елементів проблемного навчання студентів. Вони передбачають поряд із розглядом основного лекційного матеріалу встановлення та розгляд кола проблемних питань дискусійного характеру, які недостатньо розроблені в науці й мають актуальне значення для теорії та практики. Лекції проблемного характеру відрізняються поглибленою аргументацією матеріалу, що викладається. Вони сприяють формуванню у студентів самостійного творчого мислення, прищеплюють їм пізнавальні навички. Студенти стають учасниками наукового пошуку та вирішення проблемних ситуацій.

Міні-лекції передбачають викладення навчального матеріалу за короткий проміжок часу й характеризуються значною ємністю, складністю логічних побудов, образів, доказів та узагальнень. Вони проводяться, як правило, як частина заняття-дослідження. Міні-лекції відрізняються від повноформатних лекцій значно меншою тривалістю. Зазвичай міні-лекції тривають 10 – 15 хвилин і використовуються для того, щоб стисло донести нову інформацію до всіх слухачів. Міні-лекції часто застосовуються як частини цілісної теми, яку бажано викладати повноформатною лекцією, щоб не втомлювати аудиторію. Тоді інформація надається по черзі кількома окремими сегментами, між якими застосовуються інші форми й методи навчання.

Семінари-дискусії передбачають обмін думками і поглядами учасників з приводу цієї теми, а також розвивають мислення, допомагають формувати погляди та переконання, виробляють вміння формулювати думки й висловлювати їх.

Робота в малих групах дає змогу структурувати практично-семінарські заняття за формою і змістом, створює можливості для участі кожного студента в роботі за темою заняття, забезпечує формування особистісних якостей та досвіду соціального спілкування.

Мозкові атаки – метод розв'язання невідкладних завдань, сутність якого полягає в тому, щоб висловити якомога більшу кількість ідей за дуже обмежений проміжок часу, обговорити і здійснити їх селекцію.

Презентації – виступи перед аудиторією, що використовуються для представлення певних досягнень, результатів роботи групи звіту про виконання індивідуальних завдань, проектних робіт. Презентації можуть бути як індивідуальними, наприклад виступ одного слухача, так і колективними, тобто виступи двох та більше слухачів.

Метод Дельфі використовується з метою досягнення консенсусу в експертних оцінках і передбачає надання можливості висловити свої думки групі експертів, що працюють індивідуально в різних місцях. Під час вибору управлінського рішення за цим методом академічну групу розділяють, наприклад, на п'ять малих груп. Чотири групи є робочими, вони розробляють і приймають управлінське рішення, а п'ята група є експертною. Аналіз та варіанти управлінських рішень робочих груп усереднюються цією групою. Експертна група може бути поділена за спеціалізаціями.

Метод сценаріїв полягає в розробленні ймовірних моделей поведінки та розвитку конкретних явищ у перспективі.

Модерація – це метод, який допомагає групам розглядати теми, проблеми, задачі зосереджуючись на змісті цілеспрямовано і ефективно під час самостійної участі кожного у вільній колегіальній атмосфері. Модерація як спосіб проведення обговорення, швидко приводить до конкретних результатів, дає можливість всім присутнім брати участь у процесі вироблення рішень, відчуваючи при цьому свою повну відповідальність за результат. Використовується під час практичних занять.

Банки візуального супроводу сприяють активізації процесу навчання за темами навчальної дисципліни за допомогою наочності.

Таблиця 8.2

Використання методик активізації процесу навчання

Тема навчальної дисципліни	Практичне застосування методик	Методики активізації процесу навчання
1	2	3
<i>Тема 1.</i> Етапи розроблення та впровадження програм. Поняття алгоритму та типові алгоритмічні структури програмування	<i>Лабораторна робота 1.</i> Розроблення графічних схем алгоритмів методом покрокової деталізації	Семінари-дискусії, презентації
<i>Тема 4.</i> Елементи алгоритмічних мов: концепція типів даних, імена, значення, покажчики, змінні, константи, операції, вирази	<i>Лабораторна робота 2.</i> Знайомство з інтегрованим середовищем розроблення програм MS Visual C++.NET Підготовка і розв'язання на ПК задач лінійного характеру	Міні-лекція, модерація

1	2	3
Тема 5. Структурне програмування: послідовність, розгалуження та цикли	Лекція. Оператори мов програмування C/C++	Проблемна лекція, Кейс-метод
Тема 7. Процедурно-орієнтоване програмування. Рекурсія	Лабораторна робота 5. Підготовка і розв'язання на ПК задач з використанням функцій і макросів	Робота в малих групах, презентація
Тема 10. Масиви	Лекція. Одновимірні масиви мов програмування C/C++	Проблемна лекція

9. Методи контролю

Система оцінювання сформованих компетентностей (див. табл. 2.1) у студентів враховує види занять, які згідно з програмою навчальної дисципліни передбачають лекційні, 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, практичних, семінарських занять і оцінюється сумою набраних балів (максимальна сума – 60 балів; мінімальна сума, що дозволяє студенту скласти іспит, – 35 балів);

модульний контроль, що проводиться 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 графіка навчального процесу.

Поточний контроль з цієї навчальної дисципліни проводиться в таких формах:

активна робота на лекційних заняттях;

активна участь у виконанні завдань на лабораторну роботу;

виконання та захист лабораторних робіт;
захист індивідуального завдання;
проведення поточного тестування;
експрес-опитування.

Модульний контроль з цієї навчальної дисципліни проводиться у формі модульної письмової контрольної роботи.

Підсумковий/семестровий контроль проводиться у формі семестрового екзамену. **Семестрові екзамени** – форма оцінювання підсумкового засвоєння студентами теоретичного та практичного матеріалу з окремої навчальної дисципліни, що проводиться як контрольний захід.

Порядок проведення поточного оцінювання знань студентів.

Оцінювання знань студента під час лабораторних занять та виконання індивідуальних завдань проводиться за накопичувальною 100-бальною системою 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;

уміння поєднувати теорію з практикою під час розв'язання задач, проведенні розрахунків під час виконання індивідуальних завдань, та завдань, винесених на розгляд в аудиторії;

логіка, структура, стиль викладу матеріалу в письмових роботах і під час виступів в аудиторії, вміння обґрунтовувати свою позицію, здійснювати узагальнення інформації та робити висновки;

арифметична правильність виконання індивідуального та комплексного розрахункового завдання.

Максимально можливий бал за конкретним завданням ставиться за умови відповідності індивідуального завдання студента або його усної відповіді всім зазначеним критеріям. Відсутність тієї або іншої складової знижує кількість балів. Під час оцінювання індивідуальних завдань увага також приділяється якості, самостійності та своєчасності здавання виконаних завдань викладачу, згідно з графіком навчального процесу. Якщо якась із вимог не буде виконана, то бали будуть знижені.

Проведення поточно-модульного контролю. Поточно-модульний контроль здійснюється два рази за семестр (один раз у рамках вивчення змістового модуля) та оцінюється за трьома складовими: практичний модульний контроль, теоретичний модульний контроль і модульний контроль виконання ІЗ (у першому змістовому модулі). Оцінка за практичну складову модульного контролю виставляється за результатами оцінювання знань студента під час захисту лабораторних робіт та ІЗ.

Теоретичний модульний контроль здійснюється у письмовій формі, у вигляді контрольних опитувань і проміжного тестового контролю – модульних письмових контрольних робіт згідно з графіком навчального процесу.

Оцінка за теоретичну складову виставляється за результатами контрольних опитувань і тестового контролю.

Оцінка за ІЗ виставляється за результатами оцінювання знань студента під час захисту індивідуальних завдань і враховує своєчасність їх виконання.

Для підведення підсумків роботи студентів із змістового модуля виставляється підсумкова оцінка, яка складається з оцінок за поточно-модульний контроль.

Письмова модульна контрольна робота проводиться 2 рази за семестр та включає практичні й тестові завдання різного рівня складності відповідно до тем змістового модуля.

Критерії оцінювання позааудиторної самостійної роботи студентів. Загальними критеріями 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 оброблення, самореалізація під час лекційних та лабораторних занять.

Критеріями оцінювання є:

здатність проводити критичне та незалежне оцінювання 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 подання матеріалу;
використання методів порівняння, узагальнення понять та явищ;
оформлення роботи.

Порядок підсумкового контролю з навчальної дисципліни.

Підсумковий контроль знань та компетентностей студентів з навчальної дисципліни здійснюється на підставі проведення семестрового екзамену. Екзаменаційний білет охоплює програму дисципліни і передбачає визначення рівня знань та ступеня опанування студентами компетентностей (див. табл. 2.1).

Завданням екзамену є перевірка розуміння студентом програмного матеріалу в цілому, логіки та взаємозв'язків між окремими розділами, здатності творчого використання накопичених знань, вміння формулювати своє ставлення до певної проблеми навчальної дисципліни тощо.

В умовах реалізації компетентнісного підходу екзамен оцінює рівень засвоєння студентом компетентностей, що передбачені кваліфікаційними вимогами. Кожен екзаменаційний білет складається із 2-х практичних задач, які передбачають вирішення типових професійних завдань фахівця на робочому місці та дозволяють діагностувати рівень теоретичної підготовки студента і рівень його компетентності з навчальної дисципліни.

Екзаменаційний білет включає два евристичних завдання, які оцінюються відповідно до Тимчасового положення "Про порядок оцінювання результатів навчання студентів за накопичувальною бально-рейтинговою системою" ХНЕУ ім. С. Кузнеця.

Студент, який із поважних причин, підтверджених документально, не мав можливості брати участь у формах поточного контролю, тобто не склав змістовий модуль, має право на його відпрацювання у двотижневий термін після повернення до навчання за розпорядженням декана факультету відповідно до встановленого терміну.

Студент **не може бути допущений** до складання екзамену, якщо кількість балів, одержаних за результатами перевірки успішності під час

поточного та модульного контролю відповідно до змістового модуля впродовж семестру, в сумі не досягла 35 балів. Після екзаменаційної сесії декан факультету видає розпорядження про ліквідацію академічної заборгованості. У встановлений термін студент добирає залікові бали.

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

Результат семестрового екзамену оцінюється в балах (максимальна кількість – 40 балів, мінімальна кількість, що зараховується, – 25 балів) і пропонується у відповідній графі екзаменаційної *"Відомості обліку успішності"*.

Підсумкова оцінка з навчальної дисципліни розраховується з урахуванням балів, отриманих під час екзамену, 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. У випадку отримання менше 60 балів студент обов'язково здає залік після закінчення екзаменаційної сесії у встановлений деканом факультету термін, але не пізніше двох тижнів після початку семестру. У випадку повторного отримання менше 60 балів декан факультету призначає комісію у складі трьох викладачів на чолі із завідувачем кафедри та визначає термін перескладання заліку, після чого приймається рішення відповідно до чинного законодавства: "зараховано" – студент продовжує навчання за графіком навчального процесу, а якщо "не зараховано", тоді декан факультету пропонує студенту повторне вивчення навчальної дисципліни протягом наступного навчального періоду самостійно.

Екзаменаційний білет складається із двох евристичних завдань у вигляді задач. Для розв'язання кожної задачі потрібно розробити програму (проект) на алгоритмічній мові C/C++. Підсумкова оцінка за екзамен є сумою оцінок за кожне завдання.

Зразок екзаменаційного білета (денна форма навчання)

Форма № Н-5.05

Міністерство освіти і науки України
Харківський національний економічний університет ім. С. Кузнеця
Освітній ступінь "бакалавр"
спеціальності 122 "Комп'ютерні науки та інформаційні технології"
Семестр I
Дисципліна: "Програмування"

ЕКЗАМЕНАЦІЙНИЙ БІЛЕТ №

Загальні вимоги: дві задачі в одному проекті; наявність меню; багатофайловий проект.

Задача 1.

Дана прямокутна матриця цілих чисел. Визначити суму елементів у тих рядках, які містять хоча б один від'ємний елемент, та номер рядка і стовпчика максимального елемента матриці.

Вивід даних виконати за допомогою функції.

Задача 2.

В одновимірному масиві, що складається з n дійсних елементів, обчислити:

- 1) суму елементів масиву з непарними номерами;
- 2) суму елементів масиву, розташованих між першими й останнім від'ємними елементами.

Доступ до елементів масиву виконати за допомогою покажчика.

Введення елементів масиву з клавіатури та оброблення цього масиву виконати за допомогою функцій.

Розроблений проект зберегти в окремій папці на диску.

Затверджено на засіданні каф. інформаційних систем ХНЕУ ім. С. Кузнеця
Протокол № від 20 р.

Зав. кафедри

(підпис) (П.І.Б.)

Екзаменатор

(підпис) (П.І.Б.)

Зразок екзаменаційного білета (заочна форма навчання)

Форма № Н-5.05

Міністерство освіти і науки України
Харківський національний економічний університет ім. С. Кузнеця
Освітній ступінь "бакалавр"
спеціальності 122 "Комп'ютерні науки та інформаційні технології"
Семестр I
Дисципліна: "Програмування"

ЕКЗАМЕНАЦІЙНИЙ БІЛЕТ №

Загальні вимоги: наявність меню.

Задача 1.

Дана прямокутна матриця цілих чисел. Визначити суму елементів у тих рядках, які містять хоча б один від'ємний елемент, та номер рядка і стовпчика максимального елемента матриці.

Задача 2.

В одновимірному масиві, що складається з n дійсних елементів, обчислити:

- 1) суму елементів масиву з непарними номерами;
- 2) суму елементів масиву, розташованих між першими й останнім від'ємними елементами.

Розроблений проект зберегти в окремій папці на диску.

Затверджено на засіданні каф. інформаційних систем ХНЕУ ім. С. Кузнеця
Протокол № від 20 р.

Зав. кафедри

(підпис) (П.І.Б.)

Екзаменатор

(підпис) (П.І.Б.)

Підсумкові бали за екзамен: підсумкова оцінка за екзамен є сумою оцінок за кожне завдання. Кожне завдання оцінюється від 0 до 20 балів у відповідності з такою шкалою:

19 – 20 балів	Завдання виконане в повному обсязі. Програма працює правильно на всіх тестах. Інтерфейс програми задовольняє встановлені вимоги. Текст програми супроводжується коментарями
15 – 18 балів	Завдання виконане в повному обсязі Програма працює правильно. Є невеликі зауваження до організації інтерфейсу користувача або програмного коду. У тексті програми відсутні коментарі
11 – 14 балів	Завдання в основному виконано. Програма працює правильно, але не реалізована одна з функцій або можливостей, зазначених у завданні
7 – 10 балів	Завдання виконане, але не в повному обсязі. Програма працює, але не реалізовані дві або три функції або можливості, зазначені в завданні
4 – 6 балів	Завдання не виконане, але програма запускається і як мінімум дозволяє виконати одну дію
1 – 3 балів	Завдання не виконане. Програма запускається, але містить грубі помилки, у тому числі й під час організації введення вихідних даних. Під час роботи програма або зависає або аварійно перериває роботу
0 балів	Програма або взагалі не запускається або відсутня

10. Розподіл балів, які отримують студенти

Система оцінювання рівня сформованості професійних компетентностей студентів денної форми навчання наведена в табл. 10.1.

**Система оцінювання рівня сформованості
професійних компетентностей**

Професійні компетентності	Навчальний тиждень	Години	Форми навчання	Оцінка рівня сформованості компетентностей			
				Форми контролю	Макс. бал		
1	2	3	4	5	6		
Змістовий модуль 1. Організація програм							
Здатність використовувати понятійний апарат і теоретичні знання в галузі інформаційних технологій, визначати основні підходи до опису алгоритмів	2	Ауд.	2	Лекція	Тема 1. Етапи розроблення та впровадження програм. Поняття алгоритму та типові алгоритмічні структури програмування	Робота на лекції	0,5
			2	Лабораторне заняття	Лабораторна робота 1. Розроблення графічних схем алгоритмів методом покрокової деталізації	Активна участь у виконанні лабораторної роботи	0,5
						Експрес-опитування за темою 1	0,5
		СРС	11	Підготовка до занять	Пошук, підбір та огляд літературних джерел за заданою тематикою	За результатами експрес-опитувань, контрольних та лабораторних робіт в межах аудиторних занять	
Визначення завдання на лабораторний практикум, ІНДЗ та знайомство з предметною областю							
Здатність застосовувати сучасні програмні продукти, ІТ-технології та технологічні засоби у професійній діяльності; здатність до підвищення особистого професійного рівня	3	Ауд.	2	Лекція	Тема 2. Архітектура комп'ютерів, принципи Джона фон Неймана. Тема 3. Позиційні системи числення. Тема 4. Елементи алгоритмічних мов: концепція типів даних, імена, значення, покажчики, змінні, константи, операції, вирази	Робота на лекції	0,5

Продовження табл. 10.1

1	2	3	4	5	6		
		СРС	2	Лабораторне заняття Лабораторна робота 2. Знайомство з інтегрованим середовищем розроблення програм MS Visual C++.NET. Підготовка і розв'язання на ПК задач лінійного характеру	Активна участь у виконанні лабораторної роботи	0,5	
			11	Підготовка до занять Пошук, підбір та огляд літературних джерел за заданою тематикою Підготовка до експрес-опитування. Підготовка ІЗ	За результатами експрес-опитувань, контрольних та лабораторних робіт в межах аудиторних занять		
Здатність самостійно оволодівати новітніми методами, засобами, інструментами проектування інформаційних систем	4	Ауд.	2	Лекція Тема 5. Структурне програмування: послідовність, розгалуження та цикли	Робота на лекції	0,5	
			2	Лабораторне заняття Лабораторна робота 2. Знайомство з інтегрованим середовищем розроблення програм MS Visual C++.NET. Підготовка і розв'язання на ПК задач лінійного характеру	Експрес-опитування за темами 2 – 4 Активна участь у виконанні лабораторної роботи	0,5 0,5	
		СРС	6	Підготовка до занять	Пошук, підбір та огляд літературних джерел за заданою тематикою	За результатами експрес-опитувань, контрольних та лабораторних робіт в межах аудиторних занять	
					Підготовка до захисту лабораторної роботи 2. Підготовка ІЗ		
				Захист лабораторної роботи № 2	2,5		

Продовження табл. 10.1

1	2	3	4	5	6		
Здатність розробляти прикладні програми у певній предметній області	5	Ауд.	2	Лекція	Тема 6. Передпроцесорна обробка	Робота на лекції	0,5
			2	Лабораторне заняття	Лабораторна робота 3. Підготовка і розв'язання на ПК задач з розгалуженням	Активна участь у виконанні лабораторної роботи	0,5
		СРС	4	Підготовка до занять	Пошук, підбір та огляд літературних джерел за заданою тематикою	За результатами експрес-опитувань, контрольних та лабораторних робіт в межах аудиторних занять	
					Підготовка до експрес-опитування. Підготовка ІЗ		
Здатність розробляти прикладні програми у певній предметній області	6	Ауд.	2	Лекція	Тема 7. Процедурно-орієнтоване програмування. Рекурсія	Робота на лекції	0,5
			2	Лабораторне заняття	Лабораторна робота 3. Підготовка і розв'язання на ПК задач з розгалуженням	Експрес-опитування за темою 4	0,5
						Активна участь у виконанні лабораторної роботи	0,5
						Захист лабораторної роботи № 3	2,5
		СРС	6	Підготовка до занять	Пошук, підбір та огляд літературних джерел за заданою тематикою	За результатами експрес-опитувань, контрольних та лабораторних робіт в межах аудиторних занять	
Підготовка до захисту лабораторної роботи 3. Підготовка ІЗ							

Продовження табл. 10.1

1	2	3	4	5	6		
Здатність до аналізу та синтезу науково-технічної, природничо-наукової та загальнонаукової інформації	7	Ауд.	2	Лекція	Тема 8. Бібліотеки динамічного компонування (DLL)	Робота на лекції	0,5
			2	Лабораторне заняття	Лабораторна робота 4. Підготовка і розв'язання на ПК задач з використанням циклів	Активна участь у виконанні лабораторної роботи	0,5
		СРС	11	Підготовка до занять	Пошук, підбір та огляд літературних джерел за заданою тематикою	За результатами експрес-опитувань, контрольних та лабораторних робіт в межах аудиторних занять	
					Підготовка до експрес-опитування		
Підготовка ІЗ							
Виконання завдання лабораторної роботи 4							
Здатність до аналізу та синтезу науково-технічної, природничо-наукової та загальнонаукової інформації	8	Ауд.	2	Лекція	Тема 9. Методології розроблення програм: низхідне та висхідне проектування, модульне програмування	Робота на лекції	0,5
			2	Лабораторне заняття	Лабораторна робота 4. Підготовка і розв'язання на ПК задач з використанням циклів	Захист лабораторної роботи № 4	2,5
		Експрес-опитування за темою 5				0,5	
		Активна участь у виконанні лабораторної роботи				0,5	
		СРС	2	Підготовка до занять	Пошук, підбір та огляд літературних джерел за заданою тематикою	За результатами експрес-опитувань, контрольних та лабораторних робіт в межах аудиторних занять	
					Підготовка до контрольної роботи. Підготовка ІЗ		
Виконання завдання лабораторної роботи 4							

Продовження табл. 10.1

1	2	3	4		5	6			
Продемонструвати знання сучасних ІТ-технологій, мов програмування, технологічних засобів та способів їх використання в процесі виконання професійної діяльності та розв'язання професійних завдань початкового рівня складності	9	Ауд.	2	Лекція	Контрольна робота за темами 1 – 9	Активна участь у виконанні контрольної роботи	0,5		
			2	Лабораторне заняття	Лабораторна робота 5. Підготовка і розв'язання на ПК задач з використанням функцій і макросів	Контрольна робота 1	9		
						Активна участь у виконанні лабораторної роботи	0,5		
			СРС	2	Підготовка до занять	Пошук, підбір та огляд літературних джерел за заданою тематикою	За результатами експрес-опитувань, контрольних та лабораторних робіт в межах аудиторних занять		
		Підготовка до експрес-опитування							
		Підготовка до контрольної роботи							
		Підготовка до захисту ІЗ							
		Змістовий модуль 2. Основні похідні типи даних мов програмування C/C++							
		Здатність до розроблення ІТ-рішень, що спрямовані на удосконалення існуючих ІС	10	Ауд.	2	Лекція	Тема 10. Масиви	Робота на лекції	0,5
					2	Лабораторне заняття	Лабораторна робота 5. Підготовка і розв'язання на ПК задач з використанням функцій і макросів	Активна участь у виконанні лабораторної роботи	0,5
СРС	2			Підготовка до занять	Пошук, підбір та огляд літературних джерел за заданою тематикою	За результатами експрес-опитувань, контрольних та лабораторних робіт в межах аудиторних занять			
					Підготовка до експрес-опитування				
					Виконання завдання лабораторної роботи 5				

1	2	3	4	5	6			
Здатність до розроблення IT-рішень, що спрямовані на удосконалення існуючих ІС	11	Ауд.	2	Лекція	Тема 10. Масиви	Робота на лекції	0,5	
			2	Лабораторне заняття	Лабораторна робота 5. Підготовка і розв'язання на ПК задач з використанням макросів функцій і макросів	Активна участь у виконанні лабораторної роботи	0,5	
						Експрес-опитування за темами 6 – 8	0,5	
						Захист лабораторної роботи № 5	2,5	
		СРС	2	Підготовка до занять	Пошук, підбір та огляд літературних джерел за заданою тематикою	За результатами експрес-опитувань, контрольних та лабораторних робіт в межах аудиторних занять		
					Підготовка до експрес-опитування			
					Підготовка до захисту лабораторної роботи 5			
		12	Ауд.	2	Лекція	Тема 10. Масиви	Робота на лекції	0,5
				2	Лабораторне заняття	Лабораторна робота 6. Підготовка і розв'язання на ПК задач обробки однорічних масивів	Активна участь у виконанні лабораторної роботи	0,5
			СРС	2	Підготовка до занять	Пошук, підбір та огляд літературних джерел за заданою тематикою	За результатами експрес-опитувань, контрольних та лабораторних робіт в межах аудиторних занять	
Виконання завдання лабораторної роботи 6								

Продовження табл. 10.1

1	2	3	4	5	6		
Здатність застосовувати сучасні програмні продукти, IT-технології та технологічні засоби у професійній діяльності; розробляти прикладні програми у певній предметній області	13	Ауд.	2	Лекція	Тема 11. Похідні типи даних. Рядки в стилі С	Робота на лекції	0,5
			2	Лабораторне заняття	Лабораторна робота 6. Підготовка і розв'язання на ПК задач обробки одновиірних масивів	Експрес-опитування за темами 9, 10	0,5
						Захист лабораторної роботи № 6	2,5
			Активна участь у виконанні лабораторної роботи	0,5			
		СРС	2	Підготовка до занять	Пошук, підбір та огляд літературних джерел за заданою тематикою	За результатами експрес-опитувань, контрольних та лабораторних робіт в межах аудиторних занять	
					Підготовка до експрес-опитування		
Підготовка до захисту лабораторної роботи 6							
Здатність застосовувати сучасні програмні продукти, IT-технології та технологічні засоби у професійній діяльності; розробляти прикладні програми у певній предметній області	14	Ауд.	2	Лекція	Тема 11. Похідні типи даних. Рядки в стилі С	Робота на лекції	0,5
			2	Лабораторне заняття	Лабораторна робота 7. Підготовка і розв'язання на ПК задач обробки двовиірних масивів	Активна участь у виконанні лабораторної роботи	0,5
		СРС	2	Підготовка до занять	Пошук, підбір та огляд літературних джерел за заданою тематикою	За результатами експрес-опитувань, контрольних та лабораторних робіт в межах аудиторних занять	
					Підготовка до контрольної роботи		
					Виконання завдання лабораторної роботи 7		

Продовження табл. 10.1

1	2	3	4	5	6				
Здатність застосовувати сучасні програмні продукти, IT-технології та технологічні засоби у професійній діяльності; розробляти прикладні програми у певній предметній області	15	Ауд.	2	Лекція	Тема 11. Похідні типи даних. Рядки в стилі С	Робота на лекції	0,5		
			2	Лабораторне заняття	Лабораторна робота 7. Підготовка і розв'язання на ПК задач обробки двовимірних масивів	Експрес-опитування за темою 10	0,5		
						Захист лабораторної роботи № 7	2,5		
						Активна участь у виконанні лабораторної роботи	0,5		
		СРС	2	Підготовка до занять	Пошук, підбір та огляд літературних джерел за заданою тематикою	За результатами експрес-опитувань, контрольних та лабораторних робіт в межах аудиторних занять			
					Підготовка до експрес-опитування				
					Підготовка до контрольної роботи				
					Підготовка до захисту лабораторної роботи 7				
		Здатність адаптувати власну модель поведінки до умов соціокультурного середовища; здатність самостійно оволодівати новітніми методами, засобами, інструментами проектування інформаційних систем	16	Ауд.	2	Лекція	Тема 12. Введення в систему вводу-виводу С/С++	Робота на лекції	0,5
					2	Лабораторне заняття	Лабораторна робота 8. Підготовка і розв'язання на ПК задач обробки масивів з використанням покажчиків	Активна участь у виконанні лабораторної роботи	0,5
СРС	5			Підготовка до занять	Пошук, підбір та огляд літературних джерел за заданою тематикою	За результатами експрес-опитувань, контрольних та лабораторних робіт в межах аудиторних занять			
					Підготовка до контрольної роботи				
					Виконання завдання лабораторної роботи 8				

Закінчення табл. 10.1

1	2	3	4	5	6			
Продемонструвати знання сучасних IT-технологій, мов програмування, технологічних засобів та способів їх використання в процесі виконання професійної діяльності та розв'язання професійних завдань початкового рівня складності; відповідальність за прийняття та реалізацію індивідуальних та колективних рішень	17	Ауд.	2	Лекція	Контрольна робота за темами 10 – 12	Активна участь у виконанні контрольної роботи	0,5	
			2	Лабораторне заняття	Лабораторна робота 8. Підготовка і розв'язання на ПК задач обробки масивів з використанням покажчиків	Контрольна робота 2	9	
						Експрес-опитування за темами 11, 12	0,5	
						Активна участь у виконанні лабораторної роботи	0,5	
		СРС	2	Підготовка до занять	Пошук, підбір та огляд літературних джерел за заданою тематикою	За результатами експрес-опитувань, контрольних та лабораторних робіт в межах аудиторних занять		
					Підготовка до контрольної роботи			
					Підготовка до захисту лабораторної роботи 8			
		СЕСІЯ	Ауд.	2	Передекзаменаційна консультація	Вирішення практичних завдань на різні теми, що входять до підсумкового контролю	Підсумковий контроль	40
				3	Екзамен	Виконання завдань екзаменаційного білета		
			СРС	8	Підготовка до екзамену	Повторення матеріалів змістовних модулів		
Усього годин		150	Загальна максимальна кількість балів із дисципліни		100			
аудиторні		64*	поточний контроль		60			
самостійна робота		86	підсумковий контроль		40			

*без урахування екзамену та передекзаменаційних консультацій.

Розподіл балів у межах тем змістових модулів наведено в табл. 10.2.

Таблиця 10.2

Розподіл балів за темами

Поточне тестування та самостійна робота												Іспит	Сума
Змістовий модуль 1									Змістовий модуль 2				
T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12		
6,2	0,2	0,6	6,5	10	4,5	4,5	1,75	1,15	13	9	2,6	40	100

Примітка. T1, T2 ... T12 – теми змістових модулів.

Максимальну кількість балів, яку може накопичити студент протягом тижня за формами та методами навчання, наведено в табл. 10.3.

Таблиця 10.3

Розподіл балів за тижнями

Теми змістового модуля		Лекційні заняття	Практичні заняття	Лабораторні заняття	Захист лабораторних робіт	Експрес-опитування	Тестування	Письмова контрольна робота	Захист ІЗ	Усього	
Змістовий модуль 1 Організація програм	1 тиждень										
	Тема 1,2	2 тиждень	0,5	–	0,5	2,5	0,5	–	–	2,5	
	Тема 3,4	3 тиждень	0,5	–	0,5	–	–	–	–	5,5	
	Тема 5	4 тиждень	0,5	–	0,5	2,5	0,5	–	–	2,5	
	Тема 6	5 тиждень	0,5	–	0,5	–	–	–	–	4,0	
	Тема 7	6 тиждень	0,5	–	0,5	2,5	0,5	–	–	3,0	
	Тема 8	7 тиждень	0,5	–	0,5	–	–	–	–	4,0	
	Тема 9	8 тиждень	0,5	–	0,5	2,5	0,5	–	–	7,0	
Тема 1 – 9	9 тиждень	0,5	–	0,5	–	–	–	9	2	2	
Змістовий модуль 2 Основні похідні типи даних мов програмування C/C++	Тема 10	10 тиждень	0,5	–	0,5	–	–	–	–	2	
		11 тиждень	0,5	–	0,5	2,5	0,5	–	–	4	
		12 тиждень	0,5	–	0,5	–	–	–	–	1,5	
	Тема 11	13 тиждень	0,5	–	0,5	2,5	0,5	–	–	2,5	
		14 тиждень	0,5	–	0,5	–	–	–	–	5	
	15 тиждень	0,5	–	0,5	2,5	0,5	–	–	–	2,5	
	Тема 12	16 тиждень	0,5	–	0,5	–	–	–	–	1,5	
Тема 10 – 12	17 тиждень	0,5	–	0,5	2,5	0,5	–	9	–	8,5	
Разом			8	–	8	20	4	–	18	2	60

Підсумкова оцінка з навчальної дисципліни визначається відповідно до Тимчасового положення "Про порядок оцінювання результатів навчання студентів за накопичувальною бально-рейтинговою системою" ХНЕУ ім. С. Кузнеця (табл. 10.4).

Таблиця 10.4

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		

Оцінки за цією шкалою заносяться до відомостей обліку успішності, індивідуального навчального плану студента та іншої академічної документації.

11. Рекомендована література

11.1. 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. C++. 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 С++ 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.

11.2. 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 С++.NET / Д. Шеферд ; пер. с англ. – Москва : Издательско-торговый дом "Русская редакция", 2003. – 928 с. : ил.

11.3. 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.
27. REALCODING – для настоящих программистов [Электронный ресурс]. – Режим доступа : <http://www.realcoding.net/>.
28. RSDN [Электронный ресурс]. – Режим доступа : <http://www.rsdn.ru/>.

11.4. Методичне забезпечення

29. Методичні рекомендації до виконання лабораторних робіт з навчальної дисципліни "Алгоритмізація та програмування" для студентів напряму підготовки 6.050101 "Комп'ютерні науки" всіх форм навчання / В. М. Федорченко, А. В. Щербаков, Ю. Е. Парфьонов та ін. – Харків : Вид. ХНЕУ, 2012. – 180 с.
30. Методичні рекомендації до виконання лабораторних робіт з навчальної дисципліни "Основи програмування та алгоритмічні мови" для студентів напряму підготовки "Комп'ютерні науки" всіх форм навчання. Частина 2 / В. М. Федорченко, М. Ю. Лосєв, А. В. Щербаков та ін. – Харків : Вид. ХНЕУ, 2009. – 180 с.
31. Федорченко В. М. Методичні рекомендації до практичних завдань з модуля "Принципи розробки Windows-додатків" навчальної дисципліни "Основи програмування та алгоритмічні мови" для студентів напряму підготовки "Комп'ютерні науки" всіх форм навчання / В. М. Федорченко, Д. В. Гриньов, О. В. Тарасов. – Харків : Вид. ХНЕУ, 2011. – 156 с.
32. Федорченко В. М. Методичні рекомендації до виконання практичних завдань з навчальної дисципліни "Алгоритмізація та програмування" для студентів напряму підготовки "Комп'ютерні науки" всіх форм навчання / В. М. Федорченко, М. Ю. Лосєв, О. В. Тарасов. – Харків : Вид. ХНЕУ, 2010. – 92 с.

Додатки

Додаток А
Таблиця А.1

Структура складових професійних компетентностей з навчальної дисципліни "Програмування" за Національною рамкою кваліфікацій України

Складові компетентності, яка формується в рамках теми	Мінімальний досвід	Знання	Вміння	Комунікації	Автономність і відповідальність
1	2	3	4	5	6
Тема 1. Етапи розроблення та впровадження програм. Поняття алгоритму та типові алгоритмічні структури програмування					
Визначати способи формалізації і реалізації алгоритмів та програм	Сутність поняття алгоритму. Основні етапи розроблення та впровадження програм	Знання характеристик ПЗ, основних форм реалізації ПЗ; параметрів та умов забезпечення якості й ефективності	Розробляти блок-схеми алгоритмів	Ефективно формувати комунікаційну стратегію щодо розроблення та реалізації моделей та алгоритмів	Відповідальність за точну формалізацію задачі, визначення вхідних/вихідних даних; розроблення та реалізацію алгоритму
Тема 2. Архітектура комп'ютерів, принципи Джона фон Неймана					
Визначати технологію розроблення та реалізації ПЗ	Основні принципи побудови комп'ютерів	Знання технології прийняття рішень, змісту основних моделей та засобів прийняття рішень щодо розроблення ПЗ	Ідентифікувати ключові проблеми щодо розроблення ПЗ	Ефективно формувати комунікаційну стратегію щодо розроблення та реалізації моделей та алгоритмів	Відповідальність за точність і коректність прийнятого рішення та строки його реалізації

1	2	3	4	5	6
Тема 3. Позиційні системи числення					
Визначати технологію розроблення та реалізації ПЗ	Характеристика позиційних систем числення	Знання технології прийняття рішень, змісту основних моделей та засобів прийняття рішень щодо розроблення ПЗ	Ідентифікувати ключові проблеми щодо розроблення ПЗ	Ефективно формувати комунікаційну стратегію щодо розроблення та реалізації моделей та алгоритмів	Самостійно приймати ефективні рішення щодо розроблення ПЗ та відповідати за коректність і адекватність розроблених моделей
Тема 4. Елементи алгоритмічних мов: концепція типів даних, імена, значення, покажчики, змінні, константи, операції, вирази					
Визначати технологію розроблення та реалізації ПЗ	Синтаксис та семантика мов програмування C/C++	Знання технології прийняття рішень, змісту основних моделей та засобів прийняття рішень щодо розроблення ПЗ	Розробляти лінійні програми на мові C/C++	Ефективно формувати комунікаційну стратегію щодо розроблення та реалізації ПЗ	Самостійно приймати ефективні рішення щодо розроблення ПЗ та відповідати за коректність і адекватність розроблених програм
Тема 5. Структурне програмування: послідовність, розгалуження та цикли					
Визначати технологію розроблення та реалізації ПЗ	Основні оператори управління програмою	Знання технології прийняття рішень, змісту основних моделей та засобів прийняття рішень щодо розроблення ПЗ	Розробляти розгалужені та циклічні програми на мові C/C++	Ефективно формувати комунікаційну стратегію щодо розроблення та реалізації ПЗ	Самостійно приймати ефективні рішення щодо розроблення ПЗ та відповідати за коректність і адекватність розроблених програм

1	2	3	4	5	6
Тема 6. Передпроцесорна обробка					
Визначати технологію розроблення та реалізації ПЗ	Основні етапи роботи компілятора	Знання підходів до управління препроцесором	Використовувати директиви препроцесора під час розроблення програм на мові C/C++	Ефективно формувати комунікаційну стратегію щодо розроблення та реалізації ПЗ	Самостійно приймати ефективні рішення щодо розроблення ПЗ та відповідати за коректність і адекватність розроблених програм
Тема 7. Процедурно-орієнтоване програмування. Рекурсія					
Визначати технологію розроблення та реалізації ПЗ	Принципи структурного програмування	Знання методичних основ використання функцій в програмах на мові C/C++	Створювати та використовувати функції під час розроблення програм на мові C/C++	Ефективно формувати комунікаційну стратегію щодо розроблення та реалізації ПЗ	Самостійно приймати ефективні рішення щодо розроблення ПЗ та відповідати за коректність і адекватність розроблених програм
Тема 8. Бібліотеки динамічного компонування (DLL).					
Визначати технологію розроблення та реалізації ПЗ	Принципи структурного програмування та організації програм	Знання основних принципів створення та використання DLL	Створювати та використовувати DLL під час розроблення програм на мові C/C++	Ефективно формувати комунікаційну стратегію щодо розроблення та реалізації ПЗ	Самостійно приймати ефективні рішення щодо розроблення ПЗ та відповідати за коректність і адекватність розроблених програм

1	2	3	4	5	6
Тема 9. Методології розроблення програм: низхідне та висхідне проектування, модульне програмування					
Ідентифікувати ризики під час розроблення ПЗ	Основні прийоми розроблення ПЗ	Знання принципів розроблення ПЗ	Ідентифікувати ключові проблеми щодо розроблення ПЗ	Ефективно формувати комунікаційну стратегію	Відповідальність за точність і коректність прийнятого рішення
Тема 10. Масиви					
Визначати технологію розроблення та реалізації ПЗ	Сутність та операції з об'єктами типу масив	Знання методичних основ роботи з масивами в мові C/C++	Оголошувати та обробляти масиви даних під час розроблення програм на мові C/C++	Ефективно формувати комунікаційну стратегію щодо розроблення та реалізації ПЗ	Самостійно приймати ефективні рішення щодо розроблення ПЗ та відповідати за коректність і адекватність розроблених програм
Тема 11. Похідні типи даних. Рядки в стилі C					
Визначати технологію розроблення та реалізації ПЗ	Сутність та операції з об'єктами типу посилання та вказівники	Знання методичних основ роботи з вказівниками та посиланнями в мові C/C++	Оголошувати та використовувати вказівники та посилання під час розроблення програм на мові C/C++	Ефективно формувати комунікаційну стратегію щодо розроблення та реалізації ПЗ	Самостійно приймати ефективні рішення щодо розроблення ПЗ та відповідати за коректність і адекватність розроблених програм
Тема 12. Введення в систему вводу-виводу C/C++					
Визначати технологію розроблення та реалізації ПЗ	Основні засоби форматизованого виводу	Знання методичних основ використання засобів форматизованого виводу в мові C/C++	Організовувати інтерфейс користувача під час розроблення програм на мові C/C++	Ефективно формувати комунікаційну стратегію щодо розроблення та реалізації ПЗ	Відповідальність за точність і коректність прийнятого рішення

Зміст

Вступ.....	3
1. Опис навчальної дисципліни	4
2. Мета та завдання навчальної дисципліни	5
3. Програма навчальної дисципліни	8
4. Структура навчальної дисципліни.....	12
5. Теми лабораторних занять.....	13
6. Самостійна робота.....	18
6.1. Індивідуальне завдання.....	21
6.2. Контрольні запитання для самоперевірки	24
7. Індивідуально-консультативна робота	26
8. Методи навчання	26
9. Методи контролю	30
10. Розподіл балів, які отримують студенти	37
11. Рекомендована література.....	48
11.1. Основна	48
11.2. Додаткова	49
11.3. Інформаційні ресурси.....	50
11.4. Методичне забезпечення	50
Додатки.....	51

НАВЧАЛЬНЕ ВИДАННЯ

ПРОГРАМУВАННЯ

**Робоча програма
для студентів спеціальності
122 "Комп'ютерні науки та інформаційні технології"
першого (бакалаврського) рівня**

Самостійне електронне текстове мережеве видання

Укладачі: **Федорченко** Володимир Миколайович
Щербаков Олександр Всеволодович
Парфьонов Юрій Едуардович

Відповідальний за видання *О. Г. Руденко*

Редактор *В. Ю. Степаненко*

Коректор *В. Ю. Степаненко*

План 2017 р. Поз. № 99 ЕВ. Обсяг 56 с.

Видавець і виготовлювач – ХНЕУ ім. С. Кузнеця, 61166, м. Харків, просп. Науки, 9-А

*Свідоцтво про внесення суб'єкта видавничої справи до Державного реєстру
ДК № 4853 від 20.02.2015 р.*