

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ
УНІВЕРСИТЕТ

Робоча програма
навчальної дисципліни
"Алгоритмізація та програмування"
для студентів напряму підготовки "Комп'ютерні науки"
всіх форм навчання

Укладачі:

Парфьонов Ю.Е.,

Тарасов О.В.,

Федорченко В.М.,

Щербаков О.В.

Відповідальний за випуск

Пономаренко В.С.

Харків, ХНЕУ, 2011

Затверджено на засіданні кафедри інформаційних систем.
Протокол № від 07 жовтня 2010 р.

P78 Робоча програма навчальної дисципліни "Алгоритмізація та програмування" для студентів напряму підготовки "Комп'ютерні науки" усіх форм навчання. / Укл. Парфьонов Ю.Е., Тарасов О.В., Федорченко В.М., Щербаков О.В. – Харків: Вид. ХНЕУ, 2011. – 50 с. (Укр. мов.)

Наведені кваліфікаційні вимоги до студентів в галузі інформаційних систем в промисловості і у банківській сфері; тематичний план навчальної дисципліни; зміст дисципліни за модулями та темами; плани лекцій; плани практичних та лабораторних занять; самостійна робота студента; індивідуально-консультативна робота; методики активізації процесу навчання; система поточного та підсумкового контролю знань студентів.

Рекомендовано для студентів напрямку підготовки "Комп'ютерні науки".

© Харківський національний
економічний університет, 2010

© Парфьонов Ю.Е.,
Тарасов О.В.,
Федорченко В.М.,
Щербаков О.В.

Вступ

Навчальну дисципліну "Алгоритмізація та програмування" віднесено до групи освітньо-професійних дисциплін підготовки бакалаврів за спеціалізаціями "Інформаційні управляючі системи і технології", "Комп'ютерний еколого-економічний моніторинг".

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

Навчальна дисципліна є інструментальною основою для виконання аналітичної частини подальших дисциплін, а також курсових і дипломних робіт. Вона забезпечує такі дисципліни: "Об'єктно-орієнтоване програмування", "Моделі та структури даних", "Системне програмування і операційні системи", "Організація баз даних і знань", "Методи та засоби комп'ютерних інформаційних технологій", "WEB-програмування", "Технології програмування та створення програмних продуктів", "Кросплатформове програмування", "Мобільні технології", "Сучасні засоби програмування", "Кросплатформові та багатозвенні технології".

Мета навчальної дисципліни – викладення основних понять алгоритмізації і техніки застосування у програмуванні базових алгоритмічних структур (організація програм) і базових структур даних (організація даних).

Для досягнення мети поставлені такі **основні завдання**:

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

вивчення типових підходів до розробки і аналізу найбільш розповсюджених алгоритмів рішення економіко-математичних задач;

здійснення аналізу можливостей сучасних інструментальних середовищ розробки програм (на прикладі середовища Visual C++.NET);

визначення концепцій і вивчення основних принципів організації програм в середовищі ОС Windows.

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

Структура навчальної дисципліни наведена в табл. 1.

Таблиця 1

Структура програми навчальної дисципліни

Навчальна дисципліна: підготовка бакалаврів	Напрямок, спеціальність, освітньо-кваліфікаційний рівень	Характеристика навчальної дисципліни
Кількість кредитів, відповідних до ECTS – 10; у тому числі: змістовних модулів – 2; самостійна робота; індивідуальне науково- дослідне завдання (ІНДЗ)	Галузь знань: 0501 "Інформатика та обчислювальна техніка"	Обов'язкова. Рік підготовки: 1. Семестр: 1, 2
Кількість годин: усього – 360; за змістовними модулями: модуль 1 – 180 год.; модуль 2 – 180 год.	Напрямок підготовки "Комп'ютерні науки"	Лекції (теоретична підготовка) – 66 годин (32 + 34). Практичні заняття – 17 годин (0 + 17). Лабораторні заняття – 66 годин (32 + 34). Самостійна робота – 195 годин ((90 + 10 консульт.) + (81 + 10 консульт. + 4 іспит)). Індивідуальні заняття – 16 годин (16 + 0)
Кількість тижнів викладення навчальної дисципліни: 33 (16 + 17). Кількість годин за тиждень – 4, 5	Освітньо-кваліфікаційний рівень: бакалавр	Вид контролю: залік, іспит

У процесі навчання студенти отримують необхідні знання під час проведення аудиторних занять: лекційних, практичних та лабораторних. Велике значення в процесі вивчення та закріплення знань має самостійна робота студентів і виконання ІНДЗ. Усі ці види занять розроблені відповідно кредитно-модульної системи організації навчального процесу.

1. Кваліфікаційні вимоги до студентів

Дисципліна "Алгоритмізація та програмування" є базовою для бакалаврів напрямку підготовки "Комп'ютерні науки".

Необхідна навчальна база перед початком вивчення дисципліни: для успішного вивчення дисципліни необхідні базові знання, отримані студентами в об'ємі шкільної програми, а також поточні знання при паралельному освоєнні ними дисциплін "Вступ до комп'ютерних наук", "Вища математика", "Дискретна математика".

У результаті вивчення навчальної дисципліни студенти повинні:

знати:

основні етапи процесу проектування програмного забезпечення;

типові алгоритмічні конструкції;

принципи процедурного і структурованого програмування;

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

базові типи даних;

похідні типи даних: переліки, покажчики, посилання, масиви, структури, об'єднання;

оператори управління програмою;

команди передпроцесорної обробки;

правила роботи з функціями;

систему введення-виведення C++;

основні принципи роботи з файлами;

правила роботи із шаблонами;

принципи розробки Windows-додатків;

основу побудови програм на C++/CLI,

отримати такі компетенції:

здатність складати програми мовою C++, *забезпечуючи:*

вміння вирішувати задачі з курсу вищої математики (чисельне диференціювання та інтеграція, рішення рівнянь та інше);

здатність створювати і обробляти структури та масиви структур;

здатність проводити найпростішу обробку файлів;

вміння використовувати стандартні функції та функції користувача;

вміння використовувати основні WIN API функції для розробки Windows-додатків в ОС Windows;

здатність використовувати сучасне інструментальне програмне забезпечення;

вміння користуватися раніше складеними програмами і здійснювати супровід програм, вносити зміни в програму, виконувати відладку програм за допомогою вбудованих інструментальних засобів.

2. Тематичний план навчальної дисципліни

На початку вивчення дисципліни кожен студент має бути ознайомлений як з програмою дисципліни і формами організації навчання, так і зі структурою, змістом та обсягом кожного з її навчальних модулів, а також з усіма видами контролю та методикою оцінювання знань. Тематичний план дисципліни складається з двох змістовних модулів, кожний з яких об'єднує в собі відносно окремий самостійний блок дисципліни, який логічно пов'язує кілька навчальних елементів дисципліни за змістом та взаємозв'язками (табл. 2).

Таблиця 2

Тема	Кількість годин			
	Лекції	Лабораторні заняття	Індивідуальні заняття	Самостійна робота
1	2	3	4	5
Модуль 1. Введення в розробку і кодування алгоритмів				
Тема 1. Алгоритм як основне поняття програмування. Лексичні основи мов високого рівня	2	4	2	8
Тема 2. Алгоритмічна мова C++. Основні типи даних	2	4	2	8
Тема 3. Програмування обчислювальних процесів. Оператори управління програмою	2	6	2	12
Тема 4. Функції	4	4	2	12
Тема 5. Масиви	4	10	4	28
Тема 6. Похідні типи даних. Рядки	8	4	2	17
Тема 7. Введення в систему вводу-виводу C++	4	4*	2	6

Тема 8. Передпроцесорна обробка	2	4*	2*	5
Контрольні роботи	4	-	-	4
Всього за модуль	32	32	16	100
Модуль 2. Принципи розробки Windows-додатків				
Тема	Лекції	Лабораторні заняття	Практичні заняття	Самостійна робота
Тема 9. Структури та об'єднання	6	4	2	14
Тема 10. Шаблони	4	4	2	10
Тема 11 Програмування в середовищі Windows	18	14	10	60
Тема 12. Програмування на C++/CLI	2	4	1	7
Контрольні роботи	4	-	-	4
Усього годин за модуль	34	34	17	95
Усього годин	66	66	33(16 + 17)	195

* – виконується у весняному семестрі

3. Зміст навчальної дисципліни за модулями та темами

Модуль 1. Введення в розробку і кодування алгоритмів

Тема 1. Алгоритм як основне поняття програмування. Лексичні основи мов високого рівня

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

Початкові відомості про технологію програмування: процедурне, структуроване і об'єктно-орієнтоване програмування. Мови програмування: процедурні, аплікативні, системи правил, об'єктно-орієнтовні. Стандартизація мов та середовища проектування. Транслятори. Редактори. Компонувальники. Відладчики. Керуючі структури: оператори, вирази та підпрограми. Огляд сучасних інтегрованих систем програмування. Інтегроване середовище системи програмування Visual C++.NET. Платформа DOT.NET. Етапи розробки та впровадження програм. Вимоги до програмного коду.

Тема 2. Алгоритмічна мова C++. Основні типи даних

Стандарт ANSI. 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 представлення даних. Базові типи даних: логічний, символний, цілий, речовинний. Перетворення типів: неявні перетворення, явні перетворення типу.

Пріоритети операції. Зведена таблиця пріоритетності і асоціативності операцій.

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

Пріоритети операції. Зведена таблиця пріоритетності і асоціативності операцій.

Стандартні математичні функції.

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

Тема 3. Програмування обчислювальних процесів.

Оператори управління програмою

Загальні відомості про систему вводу-виводу даних.

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

Типи операторів. Найпростіший оператор, оператор-оголошення, оператор-визначення, оператор-вираз. Управляючі 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. Управляючі оператори в циклах: оператор break, оператор continue, оператор goto. Рекомендації з вибору циклів.

Тема 4. Функції

Загальні відомості про функції. Структура функції. Значення, параметри і аргументи, що повертаються. Оголошення функції. Прототипи функцій. Визначення функції. Виконання функції.

Локальні і глобальні змінні. Правило видимості змінних. Наведення типів аргументів функцій. Правила автоматичного (неявного) наведення 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.

Робота функцій. Розбиття пам'яті. Стек і функції.

Модифікатори функцій.

Тема 5. Масиви

Оголошення масивів. Ініціалізація масивів. Обробка одновимірних масивів даних економічного характеру. Алгоритми сортування масивів. Багатовимірні масиви. Ініціалізація багатовимірного масиву. Типові приклади обробки матриць.

Масиви як параметри функцій.

Тема 6. Похідні типи даних. Рядки

Рядки як масиви символів. Операції з рядками. Тип даних string. Ввід-вивід рядків.

Поняття покажчика, посилання. Покажчики і масиви. Адресна арифметика. Посилання. Приклади використання покажчиків і посилань.

Покажчики на функції. Посилання. Параметри функцій як посилання.

Організація пам'яті в сучасних процесорах і покажчики мови C++. Моделі пам'яті. Статичні і динамічні змінні. Оператори new і delete. Динамічні масиви. Динамічні масиви як параметри функцій. Зв'язні списки. Створення однозв'язного списку. Прохід однозв'язного списку. Включення нового елемента в існуючий список. Виключення елемента із списку.

Тема 7. 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'єкту cin.

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

Використання файлів для вводу-виводу даних. Створення файла. творення потоку. Відкриття потоку. "Приєднання" файла до потоку. Обміни з файлом за допомогою потоку. "Від'єднання" потоку від файла. Закриття файла. Знищення файла. Реалізація синхронного і асинхронного вводу-виводу засобами Win32 API.

Тема 8. Передпроцесорна обробка

Основи апарату макросів. Директиви препроцесора.

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

Модуль 2. Принципи розробки windows-додатків

Тема 9. Структури та об'єднання

Структури. Структури з бітовими полями. Вкладені структури. Доступ до елементів структур. Операції з структурами. Структури як параметри функцій.

Масиви структур. Показчики на структури. Передача по посиланню масивів структур.

Об'єднання. Операції з об'єднаннями. Переліки.

Функції роботи з датою та часом.

Тема 10. Шаблони

Основи апарату шаблонів. Шаблони функцій. Перевантаження шаблонів функцій. Шаблони функцій сортування.

Стандартна бібліотека шаблонів (STL). Призначення та склад STL. Контейнери. Робота з векторами, списками, стеками, чергами.

Тема 11. Програмування в середовищі Windows

Компоненти та завдання операційної системи (ОС). Допоміжні модулі операційної системи. Характеристики ОС. Інтерфейс прикладного програмування (API). Утиліти, системні програми, які обробляють, бібліотеки процедур.

Особливості ОС Windows. Графічні елементи вікна Windows. Елементи призначеного для користувача інтерфейсу Windows. Поняття повідомлення (формат повідомлення). Джерела отримання повідомлень в Windows-програмах. Цикл обробки повідомлень. Графічна схема типової Windows-програми.

Каркасний Windows-додаток. Клас вікон. Головна функція. Створення вікна. Створення вікон з використанням існуючих класів. Визначення структури WNDCLASS. Реєстрація класу вікон. Обробка повідомлень у віконній функції. Функції підтримки вікон. Приєднання даних класу вікна до вікна. Зміна зовнішнього вигляду вікна.

Головна функція додатка. Типи даних, які використовуються в Windows. Цикл обробки повідомлень. Джерела повідомлень. Рентабельні функції. Функції перехвату повідомлень. Черги повідомлень.

Органи управління. Редактори тексту. Смуги прокрутки.

Ресурси Windows-додатків. Структура файлів ресурсу. Підключення ресурсів до виконавчого файлу. Таблиці рядків. Ресурси, які визначаються користувачем. Створення меню. Головне меню додатка. Спливаюче (контекстне) меню додатка. Повідомлення меню. Системне меню додатка. Меню без використання ресурсів. Додавання меню до вікна. Повідомлення меню.

Акселератори. Опис акселераторів у файлах ресурсів. Підключення таблиць акселераторів.

Діалогові вікна та їх елементи. Модальні та немодальні діалогові вікна. Клавіатурний інтерфейс діалогового вікна. Динамічні діалогові вікна. Шаблони діалогового вікна.

Елементи управління діалогових вікон. Створення діалогових вікон. Класи передвизначених вікон. Списки прості та комбіновані. Нотифікаційні повідомлення та їх обробка. Стандартні діалогові вікна. Стандартні діалогові панелі. Панелі для відкриття або збереження файлів. Панелі вибору кольору. Панелі для вибору шрифтів.

Структура багатовіконного інтерфейсу. Головне вікно. Складання багатовіконного додатка. Дочірні вікна. Порядок взаємодії між вікнами. Додаткова інформація про обробку повідомлень в головному вікні.

Повернення забраних ресурсів. Особливості віконної процедури. Меню в багатовіконному додатку. Режими огляду та сортування.

Загальні відомості про контексти пристроїв. Концепція GDI. Інтерфейс графічних пристроїв. Ввід об'єктів до контексту пристроїв. Растрові зображення. Функції роботи з растровими зображеннями. Кольори та бітові образи. Формат монохромного бітового образу. Формат кольорового бітового образу. Кольорові профілі. Перетворення кольорів.

Піктограми. Створення піктограм. Функції роботи з піктограмами.

Алгоритми рисування в Windows. Інструменти рисування. Перо, кість, палітра та області для рисування. області виводу. Визначення області недійсної, відсікання та повідомлення WM_PAINT.

Рисування відрізків. Режими рисування. Прямокутник, який ограновує.

Процеси та потоки. Таблиця процесів. Атрибути процесу. Сервіси процесу. Стан процесу. Атрибути потоку. Створення та завершення потоку. Волокна. Планування процесів і потоків. Системи пакетної обробки. Інтерактивні системи. Системи дійсного часу. Система пріоритетів. Засоби міжпроцесного обміну даними. Буфери обміну. Атоми. Канали передачі даних. Роздільна пам'ять.

Тема 12. Програмування на C++/CLI

Поняття складки (Assembly). Відображення C++ на специфікацію CLS. Типи даних C++ і CLR. Директива #using та оператор using. Стандартний ввід-вивід. Керовані та некеровані типи. Типова безпека. Управління прибирання сміття. Створення керованого коду, прапор компіляції /clr.

4. План лекцій

Лекційні заняття проводяться в межах єдиної технології викладання комп'ютерних дисциплін в ХНЕУ, що практикується на кафедрі Інформаційних систем. Лекції читаються в аудиторіях, що мають достатню кількість місць, зручні столи, нормальну акустику, дошку і можливість підключення ТЗН. На лекції виноситься найбільш складний матеріал, і матеріал, недостатньо повно відображений в підручниках.

Змістовий модуль 1. Введення в розробку і кодування алгоритмів

Лекція 1. Вступ. Алгоритм як основне поняття програмування (Тема 1.)

1.1. Вступ до дисципліни.

1.2. Етапи розробки та виконання програм на ЕОМ.

1.3. Поняття алгоритму, властивості алгоритму. Типові алгоритмічні конструкції.

1.4. Лексичні основи мов високого рівня.

1.5. Платформа .NET.

Література: основна [4; 6; 7; 11; 16]; додаткова [19; 25].

Лекція 2. Вступ до С++ (Тема 2.)

2.1. Лексичні елементи мови С++. Операнд, змінна. Оператор привласнення. Оголошення та ініціалізація змінних.

2.2. Поняття типу даних. Класифікація і представлення даних. Базові типи даних. Перетворення типів.

2.3. Константні величини. Операції. Пріоритети операцій.

2.4. Структура програми С++. Класи пам'яті.

2.5. Загальні відомості про систему вводу-виводу даних.

2.6. Оператор привласнення. Стандартні функції.

Література: основна [2; 5; 6; 11; 13; 16]; додаткова [18 – 21; 23; 24].

Лекція 3. Програмування обчислювальних процесів (Тема 3.)

3.1. Умовні оператори.

3.2. Оператор циклу.

3.3. Оператор переходу.

Література: основна [2; 5; 6; 11; 13; 16]; додаткова [18 – 21; 23; 24].

Лекція 4. Функції (Тема 4.)

4.1. Загальні відомості про функції. Заголовні файли.

4.2. Оголошення та визначення функції. Структура функції.

4.3. Параметри функції. Значення, що повертаються. Локальні і глобальні змінні.

4.4. Простір імен.

Література: основна [2; 5; 6; 11; 13; 16]; додаткова [18 – 21; 23; 24].

Лекція 5. Функції (закінчення) (Тема 4.)

5.1. Стек і функції. Функції, що підставляються.

5.2. Перевантаження функцій.

5.3. Рекурсія.

5.4. Функції з змінним числом аргументів.

Література: основна [2; 5; 6; 11; 13; 16]; додаткова [18 – 21; 23; 24].

Лекція 6. Масиви (Тема 5.)

6.1. Алгоритми обробки масивів. Оголошення масивів. Ініціалізація масивів.

6.2. Обробка одновимірних масивів даних економічного характеру.

6.3. Алгоритми сортування масивів.

Література: основна [2; 5; 6; 11; 13; 16]; додаткова [18 – 21; 23; 24].

Лекція 7. Масиви (закінчення) (Тема 5.)

7.1. Багатовимірні масиви. Ініціалізація багатовимірного масиву.

7.2. Типові приклади обробки матриць.

7.3. Функції і масиви.

Література: основна [2; 5; 6; 11; 13; 16]; додаткова [18 – 21; 23; 24].

Лекція 8. Контрольна модульна робота

Література: основна [2; 5; 6; 11; 13; 16]; додаткова [18 – 21; 23; 24].

Лекція 9. Похідні типи даних. Показчики (Тема 6.)

9.1. Поняття показчика, посилання. Адресна арифметика.

9.2. Організація пам'яті. Статичні і динамічні змінні. Оператори new і delete.

9.3. Показчики і масиви.

Література: основна [2; 5; 6; 11; 13; 16]; додаткова [18 – 21; 23; 24].

Лекція 10. Похідні типи даних. Показчики (закінчення) (Тема 6.)

10.1. Показчики і функції.

10.2. Динамічні масиви як параметри функцій.

10.3. Зв'язні списки.

Література: основна [2; 5; 6; 11; 13; 16]; додаткова [18 – 24; 23; 24].

Лекція 11. Похідні типи даних. Рядки (Тема 6.)

11.1. Рядки і масиви символів.

11.2. Операції з рядками.

11.3. Рядки і покажчики.

Література: основна [3; 5; 6; 11; 13; 16]; додаткова [18 – 21; 23; 24].

Лекція 12. Похідні типи даних. Рядки (закінчення) (Тема 6.)

12.1. Тип даних string.

12.2. Обробка об'єктів типу string.

12.3. Ввод-вивід рядків.

Література: основна [2; 5; 6; 11; 13; 16]; додаткова [18 – 21; 23; 24].

Лекція 13. Система вводу-виводу C++ (Тема 7.)

13.1. Базові положення системи вводу-виводу C++.

13.2. Ввід даних за допомогою глобального об'єкту cin.

13.3. Вивід даних за допомогою глобального об'єкту cout.

13.4. Засоби форматування.

Література: основна [3; 5; 6; 11; 13; 16]; додаткова [18 – 21; 23; 24].

Лекція 14. Система вводу-виводу C++ (закінчення) (Тема 7.)

14.1. Використання файлів для вводу-виводу даних.

14.2. Файли з довільним доступом.

14.3. Реалізація синхронного і асинхронного вводу-виводу

Література: основна [5; 13; 14; 16]; додаткова [18 – 21; 23; 24].

Лекція 15. Передпроцесорна обробка (Тема 8.)

15.1. Директиви передпроцесора.

15.2. Умовна компіляція.

15.3. Основи апарату макросів.

15.4. Перевизначені макроси.

Література: основна [5; 6; 11; 13; 16]; додаткова [18 – 21; 23; 24].

Лекція 16. Контрольна модульна робота

Література: основна [2; 5; 6; 11; 13; 16]; додаткова [18 – 21; 23; 24].

Змістовий модуль 2. Принципи розробки Windows-додатків

Лекція 17. Структури (Тема 9.)

17.1. Структури.

17.2. Структури з бітовими полями.

17.3. Масиви структур.

17.4. Покажчики структур.

Література: основна [3; 5; 6; 11; 13; 16]; додаткова [18 – 21; 23; 24].

Лекція 18. Структури (продовження) (Тема 9.)

18.1. Структури як параметри функцій.

18.2. Передача за посиланням масивів структур.

18.3. Функції роботи з датою та часом.

Література: основна [3; 5; 6; 11; 13; 16]; додаткова [18 – 21; 23; 24].

Лекція 19. Структури (закінчення) (Тема 9.)

19.1. Об'єднання.

19.2. Переліки.

19.3. Операції з об'єднаннями.

Література: основна [3; 5; 6; 11; 13; 16]; додаткова [18 – 21; 23; 24].

Лекція 20. Шаблони функцій (Тема 10.)

20.1. Основи апарату шаблонів.

20.2. Перевантаження шаблонів функцій.

20.3. Шаблони функцій сортування.

Література: основна [5; 6; 11; 13; 16]; додаткова [18 – 21; 23; 24].

Лекція 21. Стандартна бібліотека шаблонів (STL) (Тема 10.)

21.1. Призначення та склад STL.

21.2. Контейнери. Операції зі списками та чергами.

Література: основна [1; 13]; додаткова [19; 20].

Лекція 22. Особливості програмування в середовищі Windows (Тема 11)

22.1. Особливості ОС Windows.

22.2. Графічні елементи вікна Windows.

22.3. Елементи призначеного для користувача інтерфейсу Windows.

22.4. Поняття повідомлення.

Література: основна [12; 15; 16]; додаткова [17; 22].

Лекція 23. Архітектура Windows-додатка (Тема 11.)

23.1. Поняття API Windows-функції.

23.2. Типи даних в Windows-додатках. Типи вікон.

23.3. Ініціалізація додатка.

23.4. Цикл обробки повідомлень.

Література: основна [12; 15; 16]; додаткова [17; 22].

Лекція 24. Механізми взаємодії додатка з користувачем (Тема 11.)

24.1. Файл ресурсів.

24.2. Меню додатка.

24.3. Повідомлення меню.

24.4. Акселератори.

Література: основна [12; 15; 16]; додаткова [17; 22].

Лекція 25. Контрольна модульна робота

Література: основна [1; 6; 11; 13; 16]; додаткова [18 – 21; 23; 24].

Лекція 26. Діалогові вікна та їх елементи (Тема 11.)

31.1. Типи діалогових вікон.

31.2. Функція діалогового вікна.

31.3. Оператори шаблону діалогового вікна.

31.4. Елементи управління діалогового вікна.

Література: основна [12, 15, 16]; додаткова [17, 22].

Лекція 27. Багатовіконний інтерфейс (Тема 11.)

32.1. Створення MDI-додатка.

32.2. Вікна MDI-додатка.

Література: основна [12, 15, 16]; додаткова [17, 22].

Лекція 28. Інтерфейс графічних пристроїв (GDI) (Тема 11.)

33.1. Загальні відомості про контексти пристроїв.

33.2. Контекст дисплея.

33.3. Контекст принтера.

33.4. Інформаційний контекст.

33.5. Контекст в пам'яті.

Література: основна [12; 15; 16]; додаткова [17; 22].

Лекція 29. Процеси та потоки (Тема 11.)

29.1. Процеси.

22.2. Потоки.

Література: основна [14; 15]; додаткова [17; 22].

Лекція 30. Процеси та потоки (закінчення) (Тема 11.)

30.1. Планування процесів та потоків.

30.2. Система пріоритетів.

30.3. Загальні відомості про синхронізацію.

Література: основна [14; 15]; додаткова [17; 22].

Лекція 31. Засоби міжпроцесного обміну даними (Тема 11.)

31.1. Буфери обміну.

31.2. Атоми.

31.3. Канали передачі даних.

31.4. Передача повідомлень між процесами.

31.5. Роздільна пам'ять.

Література: основна [14; 15]; додаткова [17; 22].

Лекція 32. Програмування на C++\CLI (Тема 12.)

32.1. C++ та платформа .NET.

32.2. Типи даних C++\CLI і CLR.

32.3. Стандартний ввід-вивід.

32.4. Управління прибиранням сміття.

32.5. Типова безпека.

32.6. Типи-значення і абстрактні типи. Наведення типів.

Література: основна [11; 14 – 16]; додаткова [17; 19; 22; 25].

Лекція 33. Контрольна модульна робота

Література: основна [10, 13, 14, 16]; додаткова [19,20, 21, 25].

5. Плани практичних, індивідуальних і лабораторних занять

5.1. План практичних та індивідуальних занять

Практичне (індивідуальне) заняття – форма навчального заняття, спрямована на закріплення студентом теоретичних знань, отриманих як на лекційних заняттях, так і в процесі самостійного вивчення матеріалу, а також під час виконання ІНДЗ.

На практичному (індивідуальному) занятті викладач організовує детальний розгляд студентами окремих теоретичних положень навчальної дисципліни та формує вміння та навички їх практичного застосування шляхом індивідуального виконання студентами відповідно сформульованих завдань.

Проведення практичного (індивідуального) заняття ґрунтується на попередньо підготовленому методичному матеріалі (тестах) для виявлення ступеня оволодіння студентами необхідними теоретичними положеннями, наборі завдань різної складності для їх розв'язання студентами на занятті. Практичні (індивідуальні) заняття також проводяться з метою підготовки матеріалів до лабораторних робіт, які потребують розробки програм, або попередніх розрахунків.

Практичне (індивідуальне) заняття включає проведення попереднього контролю знань, умінь і навичок студентів, постановку загальної проблеми викладачем та її обговорення за участю студентів, розв'язування завдань з їх обговоренням, розв'язування контрольних завдань, їх перевірку, оцінювання.

У процесі проведення практичного (індивідуального) заняття викладач організовує такі види методичної роботи зі студентами:

поточний контроль у вигляді письмових або комп'ютерних тестів;

дискусію питань лекцій або попередньо визначених тем рефератів, до яких студенти готуються заздалегідь;

вирішення поточних запропонованих завдань різного рівня складності;

перевірку домашніх завдань щодо розробки програм та алгоритмів;

захист індивідуальних науково-дослідних робіт окремих студентів;

індивідуальну роботу з окремими студентами для ретельного розбору матеріалу (питань), що викликає найбільше непорозумінь.

Практичне (індивідуальне) заняття проводяться в аудиторіях з однією академічною групою.

На кожному практичному (індивідуальному) занятті викладач оцінює підготовлені студентами виступи, активність у дискусії, вміння формулювати і відстоювати свою позицію тощо. Підсумкові оцінки за кожне практичне (індивідуальне) заняття вносяться у відповідний журнал. Отримані студентом оцінки за практичні (індивідуальні) заняття враховуються при виставленні поточної модульної (практичний модульний контроль) оцінки з навчальної дисципліни. Перелік тем практичних (індивідуальних) занять наведено у табл. 3.

Таблиця 3

Перелік тем практичних (індивідуальних) занять

Назва теми	Теми практичних (індивідуальних) занять	Кількість годин	Література
1	2	3	4
Модуль 1. Введення в розробку і кодування алгоритмів			
Тема 1. Алгоритм як основне поняття програмування. Лексичні основи мов високого рівня	1. Розробка графічних схем алгоритмів методом покрокової деталізації	2	Основна [1; 8; 9]; Додаткова [18; 20]

Продовження табл. 3

1	2	3	4
Тема 2. Алгоритмічна мова C++. Основні типи даних	2. Розробка розгалужених програм мовою C++	2	Основна [5; 8; 9]; Додаткова [18; 20; 24]
Тема 3. Програмування обчислювальних процесів. Оператори управління програмою	3. Розробка розгалужених та циклічних програм на мові C++	2	Основна [5; 8; 9]; Додаткова [18; 20; 22; 24]
Тема 4. Функції	4. Розробка програм з використанням функцій мовою C++	2	Основна [5; 8; 9]; Додаткова [18; 20; 22; 24]
Тема 5. Масиви	5. Розробка програм обробки одномірних масивів з використанням функцій мовою C++. 6. Розробка програм обробки двовимірних масивів з використанням функцій мовою C++	2 2	Основна [5; 8; 9; 11]; Додаткова [18; 20; 22; 24]
Тема 6. Похідні типи даних. Рядки	7. Розробка програм обробки масивів з використанням покажчиків	2	Основна [8; 9; 11]; Додаткова [18; 20; 22; 24]
Тема 7. Введення в систему вводу-виводу C++	8. Розробка програм обробки рядків, що зберігаються на зовнішніх носіях	2	Основна [8; 9; 11; 16]; Додаткова [18; 20; 22; 24]
Тема 8. Передпроцесорна обробка	9. Розробка програм обробки рядків, з використанням макросів	2	Основна [8; 10; 11; 15]; додаткова [17; 18; 20; 24]

1	2	3	4
Модуль 2. Принципи розробки Windows-додатків			
Тема 9. Структури та об'єднання	10. Розробка програм обробки об'єднань та масивів структур	2	Основна [6 – 8; 10; 11; 15]; додаткова [17; 18; 20; 22; 24]
Тема 10. Шаблони	11. Розробка програм роботи з контейнерами із використанням STL	2	Основна [11; 12]; додаткова [18; 20]
Тема 11. Програмування в середовищі Windows	12. Розробка каркасного Windows-додатка.	2	Основна [10 – 14; 16]; додаткова [19; 21]
	13. Розробка Windows-додатка з використанням меню користувача.	2	
	14. Розробка Windows-додатка з використанням вікон діалогу та інтерфейсу графічних пристроїв.	2	
	15. Розробка багатопоточного Windows-додатку.	2	
	16. Розробка Windows-додатків з використанням міжпроцесної взаємодії	2	
Тема 12. Програмування на C++/CLI	17. Розробка програм на C++/CLI	1	Основна [10; 15]; Додаткова [25]

5.2. План лабораторних занять

Лабораторне заняття – 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 експерименти чи досліді в навчальних лабораторіях з використанням комп'ютерної техніки.

Основною метою лабораторного заняття є практичне підтвердження окремих теоретичних положень та набуття практичних вмінь з виконання обчислювальних експериментів. Головна особливість цих занять полягає в тому, що вони об'єднують теорію з практикою, забезпечують їх єдність. Сукупність лабораторних занять дисципліни є лабораторним практикумом, що сплановане за єдиним задумом. Лабораторні заняття плануються після проведення лекцій. А при необхідності розробки 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 тем, що вивчається, заповнення роздатних матеріалів індивідуальними даними, розробка програм, таблиць та інше);

- послідовність підпорядкованих виконанню на занятті 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. В ході підготовки може бути створена заготовка звіту, що дозволить заощадити час на занятті.

Лабораторні заняття проводяться в аудиторіях ОЦ, академічна група ділиться на підгрупи.

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

На початку заняття, після оголошення теми, цільової установки і коротких вказівок щодо особливостей роботи, викладачем проводиться контроль підготовка 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, роздатного матеріалу з відпрацьованими індивідуальними питаннями, початкових даних для вирішення задач, заготовок звіту і та інше).

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.

В ході заняття студенти самостійно виконують передбачені завданнями дії, заносючи результати в звіт. На це відводиться до 85 – 90% часу заняття. Викладач здійснює контроль за роботою і надає допомогу при виникненні ускладнень, звертає увагу на складні, ключові моменти. При чому, основну увагу надається не вказівці на конкретну помилку, а методиці пошуку причин виникнення цих помилок.

Складання звіту – це відповідальний етап лабораторного заняття. При його складанні студенти розвивають навички аналізу, узагальнення і творчого осмислення результатів роботи, а також навички розробки документації до програмного продукту. Необхідно прагнути того, щоб студенти оформляли звіт про виконану роботу і пред'являли його викладачу до кінця лабораторної роботи. Цьому сприяє наявність наперед підготовленої заготівки, в яку послідовно заносяться всі необхідні дані і зроблені висновки. Звіт повинен бути представлений у вигляді електронного документа.

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

При оцінці 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. Повторна здача робіт, які не були прийняті, проводиться під час додаткових занять або під час наступних лабораторних занять.

Оцінки, отримані студентом за окремі лабораторні заняття вразовуються при виставленні поточної модульної оцінки з навчальної дисципліни.

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

вирішення поточних запропонованих індивідуальних завдань на лабораторну роботу;

перевірку завдань щодо розробки програм та алгоритмів;

захист лабораторних робіт окремих студентів.

Перелік тем лабораторних занять наведено у табл. 4.

Таблиця 4

Перелік тем лабораторних занять

Назва теми	Теми лабораторних занять	Кількість годин	Література
1	2	3	4
Модуль 1. Введення в розробку і кодування алгоритмів			
Тема 1. Алгоритм як основне поняття програмування. Лексичні основи мов високого рівня	1. Знайомство з інтегрованим середовищем розробки програм MS Visual C++.NET. Підготовка і розв'язання на ПЕОМ задач лінійного характеру	4	Основна [2; 5; 6; 8; 13; 16]; Додаткова [19 – 21]
Тема 2. Алгоритмічна мова C++. Основні типи даних	2. Підготовка і розв'язання на ПЕОМ задач з розгалуженням	4	Основна [2; 5; 6; 8; 13; 16]; Додаткова [19 – 21]
Тема 3. Програмування обчислювальних процесів. Оператори управління програмою	3. Підготовка і розв'язання на ПЕОМ задач з використанням циклів	6	Основна [2; 5; 6; 8; 13; 16]; Додаткова [18 – 21]
Тема 4. Функції	4. Підготовка і розв'язання на ПЕОМ задач з використанням функцій	4	Основна [2; 5; 6; 8; 13; 16]; Додаткова [18 – 21]
Тема 5. Масиви	5. Підготовка і розв'язання на ПЕОМ задач обробки одновимірних масивів. 6. Підготовка і розв'язання на ПЕОМ задач обробки двовимірних масивів	6 4	Основна [2; 5; 6; 8; 13; 16]; Додаткова [18 – 21]

Закінчення табл. 4

Тема 6. Похідні типи даних. Рядки	7. Підготовка і розв'язання на ПЕОМ задач обробки масивів з використанням покажчиків	4	Основна [2; 5; 6; 9; 13; 16]; Додаткова [18 – 21]
Тема 7. Введення в систему вводу-виводу С++	8. Підготовка і розв'язання на ПЕОМ задач з використанням рядків і файлів	4	Основна [3; 5; 6; 9; 13; 14; 16]; Додаткова [18 – 21]
Тема 8. Передпроцесорна обробка	9. Підготовка і розв'язання на ПЕОМ задач з використанням рядків і макросів	4	Основна [5; 6; 9; 13; 16]; Додаткова [18 – 21]
Модуль 2. Принципи розробки Windows-додатків			
Тема 9. Структури та об'єднання	10. Підготовка і рішення на ПК задач обробки масивів структур	4	Основна [5; 6; 9; 13; 16]; Додаткова [18 – 21]
Тема 10. Шаблони	11. Підготовка і рішення на ПК задач обробки масивів структур з використанням контейнерів	4	Основна [1; 9; 13]; Додаткова [18; 20]
Тема 11. Програмування в середовищі Windows	12. Розробка каркасного Windows-додатка.	2	Основна [9; 12; 14 – 16]; Додаткова [17; 22]
	13. Розробка Windows-додатка з використанням меню користувача та вікон діалогу.	6	
	14. Дослідження процесів і потоків	6	
Тема 12. Програмування на С++/CLI	15. Підготовка і розв'язання на ПЕОМ задач з використанням С++/CLI	4	Основна [9; 11; 16]; Додаткова [19; 25]

6. Індивідуальне навчально-дослідне завдання (ІНДЗ)

Індивідуальні навчально-дослідні завдання (ІНДЗ) виконуються самостійно при консультуванні викладачем протягом вивчення дисципліни у відповідності до графіку навчального процесу за рахунок часу відведеного на самостійну роботу.

ІНДЗ виконуються з метою закріплення, поглиблення і узагальнення знань, отриманих студентами за час навчання та набуття практичних навичок їх застосування при вирішенні практичних задач; підготовки до виконання курсових і дипломних робіт.

Індивідуальне навчально-дослідне завдання припускає наявність таких елементів наукового дослідження:

практичної значущості;

комплексного системного підходу до вирішення завдань дослідження;

теоретичного використання передової сучасної методології і наукових розробок;

наявність елементів творчості.

В процесі виконання ІНДЗ, разом з теоретичними знаннями і практичними навиками за фахом, студент повинен продемонструвати здібності до науково-дослідної роботи і вміння творчо мислити, навчитися вирішувати науково-прикладні актуальні задачі.

6.1 Тематика ІНДЗ:

"Алгоритмізація циклічних обчислювальних процесів".

Мета роботи: набуття практичних навичок розробки алгоритмів.

Завдання:

розробка графічної схеми алгоритму, розв'язання практичної задачі.

"Обробка матриць".

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

Основні завдання:

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

6.2 Вимоги до змісту

ІНДЗ повинне містити такі розділи.

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

Зміст. Повинен відтворювати назви розділів, параграфів тощо, які розкривають тему ІНДЗ, з зазначення 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 розробки теми ІНДЗ.

Основна частина. Складається з 3 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 мовою C++ з відповідними коментарями.

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

Висновки. У висновках викладають аналіз отриманих результатів. Далі формулюють пропозиції щодо практичного використання результатів ІНДЗ.

Список літератури. Джерела потрібно розміщувати в списку в алфавітному порядку прізвищ перших авторів або заголовків. Відомості про джерела, які включені до списку, необхідно давати згідно з вимогами державного стандарту з обов'язковим наведенням праць.

Додатки. У додатки можуть бути включені матеріали, що є копією документів, звітів, або розрахункові таблиці, узагальнюючі схеми чи діаграми. При наявності кількох додатків оформлюється окрема сторінка "Додатки", номер якої є останнім, що відноситься до обсягу ІНДЗ.

6.3 Вимоги до оформлення

ІНДЗ слід оформляти за допомогою комп'ютера на одній стороні аркуша білого паперу формату А 4 (210 x 297 мм) через 1,5 міжрядкового інтервалу згідно до вимог Державного стандарту України ДСТУ 3008-95 "Документація. Звіти в сфері науки і техніки. Структура та правила оформлення" (тридцять рядків на сторінку). Мінімальна висота шрифту основного тексту має бути не меншою ніж 8 мм. Цифри та букви необхідно писати чітко, висотою не менш 3,5 мм.

Обсяг ІНДЗ повинен становити 5 – 10 друкованих сторінок. Текст ІНДЗ необхідно писати (друкувати), залишаючи береги таких розмірів:

лівий – не менше 30 мм;

правий – не менше 10 мм;

верхній – не менше 20 мм;

нижній – не менше 20 мм.

Текст основної частини ІНДЗ поділяється на розділи і підрозділи. Заголовки структурних частин ІНДЗ "Зміст", "Вступ", "Розділ ...", "Висновки", "Список рекомендованої літератури", "Додатки" пишуть (друкують) з великої літери симетрично до тексту.

Заголовки підрозділів пишуть (друкують) маленькими літерами (крім першої великої) з абзацного відступу. Крапку в кінці заголовка не ставлять. В кінці заголовка, написаного (надрукованого) в підбір до тексту, ставиться крапка.

Усі структурні складові основної частини ІНДЗ починаються з нових сторінок, відокремлюються від наступного тексту одним порожнім рядком. Крапка після назви розділу або підрозділу не ставиться.

Нумерацію сторінок, розділів, підрозділів, рисунків, таблиць подають арабськими цифрами без знака №.

Структурні частини роботи "Зміст", "Вступ", "Висновки", "Список рекомендованої літератури" не нумерують. Номер розділу ставлять після слова "Розділ" на тому ж рядку, після номера крапку не ставлять, а потім з нового рядка пишуть (друкують) заголовок розділу.

Підрозділи нумерують у межах кожного розділу. Номер підрозділу складається з номера розділу та порядкового номера підрозділу, між якими ставлять крапку. В кінці номера підрозділу повинна стояти крапка, наприклад: "2.3." (третій підрозділ другого розділу). Потім у тому ж рядку йде заголовок підрозділу.

Ілюстрації та таблиці необхідно подавати безпосередньо після тексту, де вони згадані (посилання за зразком – "подано на рис. 3.1", "дивись у табл. 3.2" або "... (рис. 3.2)") вперше, або на наступній сторінці. Ілюстрації та таблиці, які розміщені на окремих сторінках, включають до загальної нумерації сторінок.

Ілюстрації позначають словом "Рис." та нумерують послідовно в межах розділу, за виключенням ілюстрацій, поданих у додатках.

Номер ілюстрації повинен складатися з номера розділу та порядкового номера ілюстрації, між якими ставиться крапка. Наприклад: "Рис. 1.2" (другий рисунок першого розділу). За умови наявності у роботі тільки однієї ілюстрації, цей рисунок нумерується як "Рис. 1". Номер ілюстрації, її назва та пояснювальні підписи розміщують послідовно під ілюстрацією.

Таблиці нумерують послідовно (за винятком таблиць, поданих у додатках) в межах розділу. В правому верхньому куті над відповідним заголовком таблиці розміщують напис "Таблиця" із зазначенням її номера. Номер таблиці повинен складатися з номера розділу та порядкового номеру таблиці, між якими крапка, наприклад: "Таблиця 1.2" (друга таблиця першого розділу).

При переносі частини таблиці на інший аркуш (сторінку) слово "Таблиця" та номер її вказують один раз справа над першою частиною таблиці, над іншими частинами пишуть слова "Продовження табл." та вказують номер таблиці, наприклад: "Продовження табл. 1.2".

7. Самостійна робота студентів

Необхідним елементом успішного засвоєння навчального матеріалу дисципліни є самостійна робота студентів з вітчизняною та закордонною спеціальною літературою. Самостійна робота є основним засобом оволодіння навчальним матеріалом у час, вільний від обов'язкових навчальних занять. Самостійна робота студентів передбачає поглиблене вивчення тем з використанням рекомендованої літератури, пошук інформації в Інтернеті, а також додаткову роботу в комп'ютерних класах для виконання індивідуальних завдань. Основні види самостійної роботи, які запропоновані студентам:

1. Вивчення лекційного матеріалу.
2. Робота з вивчення рекомендованої літератури.

3. Вивчення основних термінів та понять з галузі обчислювальної техніки і програмування.
4. Підготовка до лабораторних і практичних занять.

8. Контрольні запитання для самодіагностики

1. Структура типової програми мовою C++.
2. Бібліотечні файли, їх організація і використання.
3. Опишіть можливості застосування інтегрованого середовища Visual C++ .NET для розробки Windows-додатків.
4. Дайте класифікацію типів даних, способи опису їх в програмах, приклади використання.
5. Перерахуйте структури мови C++, що управляють, 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 елементів?
23. Чи можна ініціалізувати символьний масив рядком символів?
24. Чи можна привласнити масив масиву?
25. Визначник 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 узяття 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'єкту?
36. Опишіть структури даних – масиви. Поясніть: організація багатовимірних масивів. Для чого використовуються покажчики і посилання?
37. Програмні 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 елемента масиву значення типу `int`.
39. Як здійснюється організація файлової системи в C++?
40. Дайте визначення поняття "потік" в C++. Назвіть і охарактеризуйте стандартні потоки введення-виводу.
41. Опишіть ієрархію потокових класів в C++.
42. Назвіть засоби форматування введення-виводу.
43. Вкажіть потоки файлового введення-виводу.
44. Які операції використовуються для запису даних у файл і для читання з файлу?
45. У чому відмінність консольного застосування від графічного?
46. Призначення і склад головної функції `WinMain()`.

47. Призначення і склад структур WNDCLASS і MSG;
48. Перерахуйте характеристики класу вікна;
49. Опишіть алгоритм створення і показу вікна:
50. Призначення і принцип реалізації циклу обробки повідомлень.
51. Призначення і склад віконної функції.
52. Опишіть алгоритм створення кнопок в Windows-вікні.
53. Вкажіть принципи обробки повідомлень WM_CHAR, WM_LBUTTONDOWN, WM_RBUTTONDOWN і повідомлень групи WM_COMMAND.
54. Вкажіть призначення функцій MessageBox(), GetWindowText(), SetWindowText(), SendMessage().
55. Як створювати діалогові вікна за допомогою редактора ресурсів?
56. Вкажіть призначення типових елементів управління діалогових вікон (Button, Edit Control, Static Text, Radio Button, Check Box) і формати їх опису у файлах ресурсів.
57. Перерахуйте призначення функцій DialogBox(), GetDlgItemText(), GetDlgItem(), SetWindowText().
58. Сформулюйте концепцію побудови інтерфейсу графічних пристроїв (GDI).
59. У чому суть контекстного представлення пристроїв?
60. Призначення бібліотеки STL?
61. У чому особливості обробки події WM_PAINT.
62. Перерахуйте основні директиви компілятора.
63. Приведіть приклади функцій що працюють зі списками.
64. Вкажіть призначення файлів шаблонного проекту простого Windows-додатка.
65. Опишіть ресурси, використовувані в шаблонному проекті і механізми їх модифікації.
66. Назвіть атрибути об'єкта-процес.
67. Назвіть атрибути об'єкта-поток.
68. Важить рівні пріоритетів потоків та порядок їх зміни.
69. Що таке структура STARTUPINFO?
70. Класифікуйте засоби обміну між процесами.
71. Назвіть суть роботи атомів і каналів.

9. Індивідуально-консультативна робота

Індивідуально-консультативна робота здійснюється за графіком індивідуально-консультативної роботи у формі: 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);

б) за засвоєнням практичного матеріалу:

консультації індивідуальні і групові;

в) для комплексної оцінки засвоєння програмного матеріалу:

індивідуальне здавання виконаних робіт, звітів;

10. Методики активізації процесу навчання

При викладанні 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 визначаються не тільки методикою і технікою викладання, але і високою ефективністю навчального процесу, який виявляється в:

високій мотивації студентів;

закріпленні теоретичних знань на практиці;

підвищенні самосвідомості студентів;

виробленні здатності ухвалювати самостійні рішення;

виробленні здібності до колективних рішень;

виробленні здібності до соціальної інтеграції;

набутті навичок вирішення конфліктів;

розвитку здібності до компромісів.

Розподіл форм та методів активізації процесу навчання за темами навчальної дисципліни наведено у табл. 5.

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

Тема	Практичне застосування навчальних технологій
Модуль 1. Введення в розробку і кодування алгоритмів	
Тема 1. Алгоритм як основне поняття програмування. Лексичні основи мов високого рівня	<i>Міні-лекція, семінар-дискусія, мозкова атака</i> "Розробка графічних схем алгоритмів"; <i>презентація</i> результатів роботи в малих групах
Тема 2. Алгоритмічна мова C++. Основні типи даних	<i>Міні-лекція, семінар-дискусія</i> "Способи завдання логічних функцій мовою C++. "; <i>ділова гра</i> "Представлення даних у пам'яті ЕОМ"
Тема 3. Програмування обчислювальних процесів. Оператори управління програмою	<i>Проблемна лекція</i> "Організація циклічних процесів"; <i>Кейс-метод</i> "Розширення можливостей циклічних програм"
Тема 4. Функції	<i>Проблемна лекція</i> "Використання процедурного програмування"; <i>презентація</i> результатів роботи в малих групах
Тема 5. Масиви	<i>Проблемна лекція</i> з питання "Порівняння алгоритмів сортування масивів"
Тема 6. Похідні типи даних. Рядки	<i>Міні-лекція, семінар-дискусія</i> "Оптимальна обробка символічної інформації"
Тема 7. Введення в систему вводу-виводу C++	<i>Проблемна лекція</i> "Шляхи удосконалення методів зберігання даних" <i>Ознайомлювальні ігри</i> "Оцінка властивостей двійкових файлів"
Тема 8. Передпроцесорна обробка	<i>Міні-лекція, семінар-дискусія</i> "Універсальні властивості макросів"; <i>презентація</i> результатів роботи в малих групах
Модуль 2. Принципи розробки Windows-додатків	
Тема 9. Структури та об'єднання	<i>Міні-лекція, семінар-дискусія</i> "Засоби зберігання та обробки табличних даних"; <i>презентація</i> результатів роботи в малих групах
Тема 10. Шаблони	<i>Модерація</i> "Що робити з чергами?"
Тема 11. Програмування в середовищі Windows.	<i>Початкова гра</i> за темою "Події в сучасних ОС"
Тема 12. Програмування на C++/CLI	<i>Міні-лекція, семінар-дискусія</i> "Безпечне програмування" <i>презентація</i> результатів роботи в малих групах

Проблемні лекції спрямовані на розвиток логічного мислення студентів і характеризуються тим, що коло питань теми обмежується двома-трьома ключовими моментами, увага студентів концентрується на матеріалі, що не знайшов відображення в підручниках, використовується досвід закордонних навчальних закладів з роздачею студентам під час лекцій друкованого матеріалу та виділенням головних висновків з питань, що розглядаються. Під час проведення лекцій студентам даються запитання для самостійного розмірковування, проте лектор сам відповідає на них, не чекаючи відповідей студентів. Система питань в ході лекції відіграє активізуючу роль, примушує студентів сконцентруватися і почати активно мислити в пошуках правильної відповіді. Використовуються під час викладання лекцій за темами 3, 4, 5, 6.

Міні-лекції передбачають виклад навчального матеріалу за короткий проміжок часу й характеризуються значною ємністю, складністю логічних побудов, образів, доказів та узагальнень. Міні-лекції проводяться, як правило, як частина заняття-дослідження і використовуються під час практичних, лабораторних занять та консультацій.

Робота в малих групах використовується з метою активізації роботи студентів при проведенні семінарських і практичних занять. Це так звані групи психологічного комфорту, де кожен учасник відіграє свою особливу роль і певними своїми якостями доповнює інших. Використання цієї технології 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, використовується під час лабораторних занять.

Семінари-дискусії передбачають обмін думками і поглядами учасників з приводу окремої теми, а також розвивають мислення, допомагають формувати погляди і переконання, виробляють вміння формулювати думки й висловлювати їх, вчать оцінювати пропозиції інших людей, критично підходити до власних поглядів. Використовуються під час захисту лабораторних робіт.

Мозкові атаки – це метод розв'язання невідкладних завдань за дуже обмежений час. Сутність його полягає в тому, щоб висловити як найбільшу кількість ідей за невеликий проміжок часу, обговорити і здійснити їх селекцію. Використовуються під час викладання лекцій, практичних занять.

Ознайомлювальні або початкові ігри частіше за все використовуються на початку занять для створення робочої атмосфери, "настройки" учасників на групову роботу.

Презентації – виступи перед аудиторією, використовуються для представлення певних досягнень, результатів роботи групи, звіту про виконання індивідуальних завдань, інструктажу, демонстрації нових товарів і послуг.

Кейс-метод – метод аналізу конкретних ситуацій, який дає змогу наблизити процес навчання до реальної практичної діяльності спеціалістів і передбачає розгляд виробничих, управлінських та інших ситуацій, складних конфліктних випадків, проблемних ситуацій, інцидентів у процесі вивчення навчального матеріалу. Використовуються під час практичних занять.

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

11. Система поточного та підсумкового контролю знань студентів

У процесі навчання студенти отримують необхідні знання під час лекційних занять, виконуючи практичні завдання на лабораторних та практичних заняттях щодо розробки алгоритмів та програм.

Оцінювання знань, умінь та навичок студентів враховує види занять, які згідно з програмою навчальної дисципліни "Алгоритмізація та програмування" передбачають лекційні, лабораторні та практичні заняття, а також самостійну роботу та виконання індивідуальних завдань.

Перевірка та оцінювання знань студентів може проводитись кількома методами:

1. Оцінювання знань студента під час практичних занять.
2. Виконання індивідуальних навчально-дослідних завдань.
3. Виконання та захист лабораторних робіт.

4. Проведення поточно-модульного контролю.
5. Проведення заліку.
6. Проведення підсумкового письмового іспиту.

Оцінювання знань студента під час практичних, лабораторних занять та виконання індивідуальних завдань проводиться за 12-ти бальною шкалою за такими критеріями:

1) ступінь засвоєння фактичного теоретичного матеріалу навчальної дисципліни;

2) знайомлення з рекомендованою літературою, а також із сучасною літературою з питань, що розглядаються;

3) уміння поєднувати теорію з практикою при розв'язанні задач, проведенні розрахунків при виконанні індивідуальних завдань, та завдань, винесених на розгляд в аудиторії;

4) якість розробки та роботи програмних продуктів;

5) логіка, структура, стиль викладу матеріалу в письмових роботах, звітах і при виступах в аудиторії, вміння обґрунтовувати свою позицію, здійснювати узагальнення інформації та робити висновки.

Оцінка "відмінно" (10-12 балів) ставиться за умови відповідності індивідуального завдання студента, або його усної відповіді усім п'ятьом зазначеним критеріям. Відсутність тієї або іншої складової знижує оцінку на відповідну кількість балів.

При оцінюванні індивідуальних завдань увага також приділяється якості, самостійності та своєчасності здачі виконаних завдань викладачу (згідно з графіком навчального процесу). Якщо якась із вимог не буде виконана, то оцінка на розсуд викладача, буде знижена.

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

Теоретичний модульний контроль здійснюється у письмовій формі, у вигляді контрольних опитувань на кожному практичному занятті і проміжного тестового контролю - контрольних робіт згідно графіка навчального процесу. Оцінка за теоретичну складову модульного

контролю виставляється за результатами контрольних опитувань і тестового контролю.

Оцінка за ІНДЗ виставляється за результатами оцінювання знань студента при захисті індивідуальних завдань і враховує своєчасність їх виконання.

Для підведення висновків роботи студентів із першого змістовного модуля виставляється підсумкова залікова оцінка, яка складається з оцінок за поточно-модульний контроль.

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

Проведення підсумкового письмового іспиту. Умовою допуску до іспиту є позитивні оцінки поточно-модульного контролю. Підсумковий контроль знань студентів здійснюється у письмовій формі за екзаменаційними білетами за 12-ти бальною шкалою.

Екзаменаційний білет складається із двох евристичних завдань у вигляді задач. Для розв'язання кожної задачі потрібно розробити програму алгоритмічною мовою C++. Підсумкова оцінка за екзамен є сумою оцінок за кожне завдання. Кожне завдання оцінюється від 0 до 6 балів у відповідності з такою шкалою:

6 балів	Завдання виконане в повному об'ємі. Програма працює правильно на всіх тестах. Інтерфейс програми задовольняє встановлені вимоги. Текст програми супроводжується коментарями
5 балів	Завдання виконане в повному обсязі. Програма працює правильно. Є невеликі зауваження до організації інтерфейсу користувача. У тексті програми відсутні коментарі
4 бали	Завдання в основному виконано. Програма працює правильно, але не реалізована одна з функцій, або можливостей, зазначених у завданні
3 бали	Завдання виконане, але не в повному обсязі. Програма працює, але не реалізовані дві або три функції, або можливості, зазначені в завданні
2 бали	Завдання не виконане, але програма запускається і, як мінімум дозволяє виконати одну дію.
1 бал	Завдання не виконане. Програма запускається, але містить грубі помилки, у тому числі й при організації уведення вихідних даних. Під час роботи програма або зависає, або аварійно перериває роботу
0 балів	Програма або взагалі не запускається, або відсутня

Для підведення підсумків роботи студентів з навчальної дисципліни виставляється загальна оцінка, яка враховує оцінки з кожного виду контролю (чотири оцінки поточно-модульного контролю за роботу протягом двох семестрів та оцінки за результатами іспиту).

$$P_o = 0.6 \times E_o + 0.1 \times (M_1 + M_2 + M_3 + M_4) ,$$

де P_o - підсумкова оцінка з навчальної дисципліни

E_o - екзаменаційна оцінка

M_1, M_2, M_3, M_4 – оцінки за модулями.

Підсумкова оцінка з дисципліни згідно з Методикою переведення показників успішності знань студентів Університету в систему оцінювання за шкалою ECTS конвертується в підсумкову оцінку за шкалою ECTS (Табл. 6).

Таблиця 6

**Переведення показників успішності знань студентів ХНЕУ
в систему оцінювання за шкалою ECTS**

Відсоток студентів, які зазвичай досягають відповідної оцінки	Оцінка за шкалою ECT3		Оцінка за бальною шкалою, що використовується в ХНЕУ	Оцінка за національною шкалою
10	відмінне виконання	A	12 – 11	відмінно
25	вище середнього рівня	B	10	
30	взагалі робота правильна, але з певною кількістю помилок	C	9 – 7	добре
25	непогано, але зі значною кількістю недоліків	D	6	задовільно
10	виконання задовольняє мінімальним критеріям	E	5 – 4	
–	потрібне повторне перекладання	FX	3	незадовільно
–	повторне вивчення дисципліни	F	2 – 1	

Зразок екзаменаційного білета

Міністерство освіти і науки України
Харківський національний економічний університет

Дисципліна: "Алгоритмізація та програмування"

ЕКЗАМЕНАЦІЙНИЙ БІЛЕТ № 2

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.

Вихідні дані ввести з текстового файлу. Вивід даних виконати за допомогою функції.

Задача 2.

В одномірному 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.

Для розв'язання задачі використати контейнер типу список.

Уведення елементів масиву з клавіатури та обробку цього масиву виконати за допомогою функцій.

Розроблений проект зберегти в окремій папці на диску.

Затверджено на засіданні кафедри
_____ протокол № _____

інформаційних систем
_____ від _____ 20 ____ р.

Зав. кафедрою _____

(підпис) (П.І.Б.)

Екзаменатор _____

(підпис) (П.І.Б.)

12. Рекомендована література

12.1 Основна

1. Аммерааль Л. STL для програмістів на C++. : Пер. с англ. / Л. Аммерааль – М. : ДМК, 1999. – 240 с. : ил.
2. Браткевич В. В. Основы программирования и алгоритмические языки (язык программирования C++). Часть 1. Конспект лекций. / В. В. Браткевич, Ю. В. Перколаб, Л. И. Лукашева – Харьков : Изд. ХГЭУ, 2001. – 68 с.
3. Браткевич В. В. Основы программирования и алгоритмические языки (язык программирования C++). Часть 2. Конспект лекцій / В. В. Браткевич, Ю. В. Перколаб, Л. И. Лукашева – Харьков : Изд. ХГЭУ, 2002. – 152 с.
4. Вирт Н. Алгоритмы и структуры данных ; пер. с англ. / Н. Вирт – М. : Мир, 1989. – 360 с., : ил.
5. Деннис М. Язык программирования С. ; Пер. с англ. / Деннис М. Ритчи, Брайан У. Керниган. – М. : Издательский дом "Вильямс", 2009. – 304 с.
6. Ишкова Э. А. C++. Начала программирования. 3-е Изд., перераб. и доп. / Э. А. Ишкова. – М. : ООО "БИНОМ-Пресс", 2004. – 368 с. : ил.
7. Кнут Д. Искусство программирования для ЭВМ / Д. Кнут – т.1-3, М. : Мир, 1976 – 1977.
8. 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 1. / В. М. Федорченко, М. Ю. Лосєв, А. В. Щербаков, Ю. Э. Парфенов. – Харків : Вид. ХНЕУ, 2008. – 216 с.
9. 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, Ю. Э. Парфенов. – Харків : Вид. ХНЕУ, 2009. – 179 с.
10. Методичні рекомендації до виконання практичних завдань з навчальної дисципліни "Основи програмування та алгоритмічні мови" для студентів напряму підготовки "Комп'ютерні науки" всіх форм навчання / В. М. Федорченко, М. Ю. Лосєв, О. В. Тарасов. – Харків : Вид. ХНЕУ, 2010. – 179 с.
11. Пирогов В. Ю. Программирование на Visual C++.NET. / В. Ю. Пирогов. – СПб. : БХВ-Петербург, 2003. – 800 с.
12. Румянцев П. В. Азбука программирования в Win32 API. / П. В. Румянцев – М. : Горячая линия – Телеком. 2001. – 312 с.
13. Харви Дейтел Как программировать на C++ ; пер. с англ. / Харви Дейтел, Пол Дейтел. – М. : ЗАО "Издательство БИНОМ", 2008. – 1455 с.

14. Харт, Джонсон М. Системное программирование в среде Windows, 3-е изд. ; пер. с англ. / Харт, Джонсон М. – М. : Издательский дом "Вильямс", 2005. – 592 с. : ил.

15. Щупак Ю. А. Эффективная разработка приложений. / Ю. А. Щупак – СПб. : Питер, 2007. – 572 с. : ил.

16. Хортон А. Visual C++ 2005: базовый курс ; пер. с англ. / А. Хортон. – М. : Издательский дом "Вильямс", 2007. – 1152 с. : ил.

12.2. Додаткова

17. Верма Р. Д. Справочник по функциям Win32 API. / Р. Д. Верма – М. : Горячая линия – Телеком, 2002. – 488 с.

18. Марченко А. Л. С++. Бархатный путь. / А. Л. Марченко – М. : Горячая линия Телеком, 1999. – 400 с.

19. Павловская Т. А. С/С++. Программирование на языке высокого уровня. / Т. А. Павловская – СПб. : Питер, 2006. – 461 с. : ил.

20. Подбельский В. В. Язык С++ : Учебн. пособие. / В. В. Подбельский – 4-е изд. – М. : Финансы и статистика, 1999. – 560 с.

21. Подбельский В. В. Программирование на языке С : - 2-е доп. Изд. / В. В. Подбельский, С. С. Фомин – М.: Финансы и статистика, 2004. – 600 с.

22. Саймон Р. Windows 2000 API. Энциклопедия программиста; пер. с англ. / Р. Саймон – К. : ООО "ДиасофтЮП", 2002. – 1088 с.

23. Страуструп Б. Язык программирования С++, 3-е изд. Специальное издание; пер. с англ. / Б. Страуструп. – СПб.; М. : "Невский диалект" – Издательство "БИНОМ", 2004. –1104с. : ил.

24. Страуструп Б. Дизайн и эволюция языка С++: пер. с англ. / Б. Страуструп – М. : ДМК, 2000. – 444 с.

25. Шеферд Д. Программирование на Microsoft Visual C++.NET. : пер. с англ. / Д. Шеферд – М. : Издательско-торговый дом "Русская редакция", 2003. – 928с. : ил.

12.3. Ресурси мережі Internet

26. Главная страница MSDN, MSDN по-русски. – Режим доступа : <http://msdn.microsoft.com/ru-ru/default.aspx>.

27. Интернет Университет Информационных Технологий - дистанционное образование. – Режим доступа : <http://www.intuit.ru/>

28. CodeNet - Все для программиста. – Режим доступа : <http://www.codenet.ru/>.

29. С++ для начинающих. – Режим доступа : <http://valera.asf.ru/cpp/book/>.

30. MSDN-WindowsAPI. – Режим доступа : http://www.vsokovikov.narod.ru/New_MSDN_API/index_msdn.htm.

31. RSDN. – Режим доступа : <http://www.rsdn.ru/>.

32. <http://www.realcoding.net>.

Зміст

Вступ.....	3
1. Кваліфікаційні вимоги до студентів.....	5
2. Тематичний план навчальної дисципліни	6
3. Зміст навчальної дисципліни за модулями та темами	7
4. План лекцій	13
5. Плани практичних, індивідуальних та лабораторних занять	18
5.1. План практичних та індивідуальних занять.....	18
5.2. План лабораторних занять	21
6. Індивідуальне навчально-дослідне завдання (ІНДЗ)	26
6.1. Тематика ІНДЗ.....	26
6.2. Вимоги до змісту	27
6.3. Вимоги до оформлення	28
7. Самостійна робота студентів	29
8. Контрольні запитання для самодіагностики.....	30
9. Індивідуально-консультативна робота	33
10. Методики активізації процесу навчання	33
11. Система поточного та підсумкового контролю знань студентів	36
12. Рекомендована література.....	41
12.1. Основна	41
12.2. Додаткова	42
12.3. Ресурси мережі Internet	42

Навчальне видання

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ "Алгоритмізація та програмування" для студентів напряму підготовки "Комп'ютерні науки" всіх форм навчання

Укладач Федорченко В.М.

Відповідальний за Пономаренко В.С.

випуск

Редактор

Коректор

План 2010 р. Поз. № 195

Підп. до друку. Формат 60x90 1/16. Папір ТАТРА. Друк офсетний.

Ум.-друк. арк. . Обл.-вид. арк. Тир. 50 прим. Зам. № Безкоштовно.

Свідоцтво про внесення до Державного реєстру суб'єктів видавничої справи
Дк № від р.

Видавець і виготівник – видавництво ХНЕУ, 61001, м. Харків, пр. Леніна, 9а