

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ СЕМЕНА КУЗНЕЦЯ

"ЗАТВЕРДЖУЮ"

Заступник керівника

(проректор з науково-педагогічної роботи)



М. В. Афанасьєв

ПРОГРАМУВАННЯ

робоча програма навчальної дисципліни

Галузь знань
Спеціальність
Освітній рівень
Освітня програма

12 Інформаційні технології
усі
перший (бакалаврський) рівень
усі

Вид дисципліни
Мова викладання, навчання та оцінювання

базова
українська

Завідувач кафедри
інформаційних систем

Руденко Олег Григорович

Харків
ХНЕУ ім. С. Кузнеця

2019

ЗАТВЕРДЖЕНО
на засіданні кафедри інформаційних систем
Протокол № 1 від 27.08.2018 р.

Розробники:
Федорченко Володимир Миколайович, доцент кафедри інформаційних систем

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

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

1. Вступ

Анотація навчальної дисципліни: Дисципліну “Програмування” віднесено до групи освітньо-професійних дисциплін підготовки бакалаврів за спеціальностями 121 “Інженерія програмного забезпечення”, 122 “Комп’ютерні науки”, 124 “Управління складними системами”, 126 “Інформаційні системи та 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 проєктів. Вона забезпечує наступні 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”.

Мета навчальної дисципліни: засвоєння необхідних знань щодо основних понять алгоритмізації і техніки застосування у програмуванні базових алгоритмічних структур (організація програм) і базових типів даних, вивчення основних етапів процесу проектування програмного забезпечення і визначення принципів процедурного програмування щодо розроблення програм мовами C/C++, Python.

Курс	1	
Семестр	1, 2	
Кількість кредитів ECTS	10	
Аудиторні навчальні заняття	лекції	64
	лабораторні	64
Самостійна робота		172
Форма підсумкового контролю	іспит	

Структурно-логічна схема вивчення навчальної дисципліни:

Попередні дисципліни	Наступні дисципліни
Основи алгоритмізації	Об’єктно-орієнтоване програмування
Вступ до фаху	Операційні системи
Вища математика	Бази даних
	Розподілені та паралельні обчислення
	Кросплатформене програмування
	Програмування для мобільних пристроїв
	WEB-технології та WEB-дизайн

2. Компетентності та результати навчання за дисципліною:

Компетентності	Результати навчання
Визначати цільові програмні платформи	Вміння аналізувати існуючий стан предметної області, аналізувати та розробляти вимоги до програмного забезпечення (ПЗ), що створюється
Визначати інструментальні засоби розроблення ПЗ	Використовувати сучасне інструментальне програмне забезпечення; користуватися раніше складеними програмами і здійснювати супровід програм, вносити зміни в програму, виконувати владодження програм за допомогою вбудованих інструментальних засобів
Визначати оптимальні методи розроблення ПЗ	Використовувати сучасні методи та засоби для проектування програмного забезпечення та компонентів архітектури програмного продукту
Визначати способи формалізації та реалізації ПЗ	Використовувати ефективні алгоритми та структури даних для розробки програмних продуктів
Здійснювати обґрунтований вибір економіко-математичних методів та моделей для їх програмної реалізації	Застосовувати сучасні інформаційні ресурси та сервіси у процесі розв'язання професійних задач, вміти конструювати програмне забезпечення для рішень в інфокомунікаціях
Здійснювати розроблення алгоритмів та програм реалізації математичних методів та моделей	Вміння застосування типових підходів до розроблення і аналізу найбільш розповсюджених алгоритмів вирішення економіко-математичних задач
Здійснювати кодування розроблених алгоритмів на мові програмування високого рівня	Застосовувати та створювати програми на мовах програмування C/C++, Python з використанням сучасних інструментальних середовищ
Застосовувати методи відлагодження та тестування ПЗ	Здійснювати планування тестування, розробку тестових сценаріїв та процедур, використовувати підходи до розробки тестів, аналізувати особливості розробки й генерації тестів, здійснювати автоматизацію тестового циклу
Здійснювати оцінювання отриманих результатів	Використовувати свої знання для побудови математичної моделі та оцінювання коректності постановок задач, а також самостійно передбачати наслідки отриманих результатів

3. Програма навчальної дисципліни

Змістовий модуль 1. Програмування на мовах C/C++. Лексичні основи мов C/C++

Тема 1. Етапи розроблення та впровадження програм

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.

Поняття алгоритму. Властивості алгоритму. Типові алгоритмічні конструкції. Розроблення алгоритму методом покрокового уточнення.

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 C/C++: 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, символьний, 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, символьні (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. Програмування на мовах C/C++. Основи програмування на мовах 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. Введення-виведення 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 і динамічні змінні. Оператори new і delete. Динамічні масиви. Динамічні масиви як параметри функцій.

Тема 12. Структури та об'єднання. Динамічні структури даних

Структури, оголошення та визначення об'єктів типу структура. Відмінності структур мов C та C++. Методи в структурах. Статичні члени структури. Структури з бітовими полями. Вкладені структури. Доступ до елементів структур. Операції з структурами. Структури як параметри функцій. Структури як значення функції, що повертається.

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

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

Визначення, класифікація динамічних структур даних, способи оголошення, ініціалізація динамічних структур, методи доступу до даних динамічних структур, розміщення їх у пам'яті, переваги і недоліки використання динамічних структур в програмах.

Зв'язні списки, однозв'язні і двонаправлені списки, циклічні списки. Створення однозв'язного списку. Прохід однозв'язного списку. Включення нового елемента в існуючий список. Виключення елемента із списку.

Створення двонаправленого списку. Прохід двонаправленого списку. Включення нового елемента в існуючий список. Виключення елемента із списку.

Тема 13. Введення в систему вводу-виводу C/C++. Файлові структури даних

Базові положення системи вводу-виводу C/C++. Потоки і буфери. Стандартні об'єкти вводу-виводу. Ієрархія класів вводу-виводу C++.

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. Використовування функції getline().

Вивід даних за допомогою глобального об'єкту cout. Очищення буфера виведу – flush(). Використовування функцій put() і write(). Ввід – вивід даних, що форматується.

Маніпулятори вводу-виводу. Функції width(), precision(), fill(). Маніпулятори, визначувані користувачем.

Використовування файлів для вводу-виводу даних. Створення файлу. Створення потоку. Відкриття потоку. «Приєднання» файлу до потоку.

Режими відкриття та доступу до файлів. Обміни з файлом з допомогою потоку. «Від'єднання» потоку від файлу. Закриття файлу. Знищення файлу. Визначення стану потоку. Функції управління вводу-виводу.

Тема 14. Шаблони. Стандартна бібліотека шаблонів

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

Стандартна бібліотека шаблонів (STL). Призначення та склад STL. Контейнери. Алгоритми. Функціональні об'єкти. Ітератори, ітератори потоку. Робота з векторами, списками, стеками, чергами.

Тема 15. Стандартний клас string

Тип даних string. Зв'язок класу string з STL. Оголошення та визначення об'єктів типу string. Методи та властивості класу string. Операції з рядками. Функції роботи з рядками. Ввід-вивід рядків. Перетворення між string та рядками в стилі C.

Тема 16. Обробка виключень. Особливості стандартів C11, C++11, C++14

Обробка виключень в мові C, функції abort(), exit(), atexit().

Принципи обробки виключень в мові C++. Генерація виключень. Послідовність обробки виключень. Особливості обробки виключень при динамічному виділенні пам'яті.

Цикл по колекції. Списки ініціалізації, універсальна ініціалізація. Переліки зі строгою типізацією. Додаткові контейнери та зміни в контейнерах. Зовнішні шаблони. Шаблони зі змінною кількістю аргументів. Клас std::tuple. Вивід типів. Лямда-функції, альтернативний синтаксис функцій. Регулярні вирази. Багатопоточне програмування, бібліотека Thread.

Змістовий модуль 3. Програмування на мові Python . Лексичні основи мови Python

Тема 17. Синтаксис та семантика мови Python концепція типів даних, імена, значення, покажчики, змінні, константи, операції, вирази

Скриптові мови програмування. Стандарти, сфери застосування Python. Парадигми програмування, що підтримує Python. Огляд сучасних інтегрованих систем програмування для Python. Інтегроване середовище системи програмування PyCharm.

Концепція типів даних Python. Імена, значення, покажчики, змінні, операції, вирази. Коментарі, Shebang. Пріоритети операцій. Таблиця пріоритетності й асоціативності операцій. Операції зі строками. Вираз if/else. Основні функції.

Тема 18. Оператори мови Python

Оператор привласнення. Оголошення й ініціалізація змінних.

Типи операторів.

Умовна конструкція if. Цикли. Слово else. Керівні оператори в циклах: оператор break, оператор continue. Функція range. Вкладені цикли.

Тема 19. Функції в мові Python

Іменні функції, інструкція def. Аргументи функції. Функції з довільним числом аргументів, функції з позиційними і іменованими аргументами, обов'язковими і необов'язковими. Вкладені функції. Анонімні функції, інструкція lambda.

Оператор return. Функція main. Декоратори. Область видимості змінних, ключове слово global.

Тема 20. Робота з модулями

Інструкції import і from. Підключення модуля зі стандартної бібліотеки. Використання псевдонімів. Налаштування простору імен. Змінні __name__, __main__. Области видимості.

Тема 21. Правила написання і документування коду на мові Python

Зовнішній вигляд коду. Прогалини в виразах і інструкціях. Коментарі. Контроль версій. Угоди по іменуванню.

Рядки документації. Атрибут __doc__. Однорядкові та багаторядкові рядки документації.

Змістовий модуль 4. Програмування на мові Python. Основи програмування на мові Python

Тема 22. Обробка виключень в мові Python

Runtime error. Інструкція try...except. Блок finally і інструкція else. Ієрархія виключень. Отримання інформації про виключення, оператор as.

Генерація виключень, оператор raise.

Менеджери контексту, конструкція with ... as.

Тема 23. Списки, кортежі і словники

Індекси і зрізи. Перебір елементів, порівняння списків, методи і функції по роботі зі списками. Генератори і ітератори. Корутіни. Додавання і видалення елементів. Перевірка наявності елемента. Сортування, підрахунок входжень. Копіювання списків, з'єднання списків. Списки списків. Складні кортежі. Функція tuple(). Перетворення зі списку в словник. Отримання і зміна елементів словника. Функція dict(). Метод get. Оператор del, метод pop(). Копіювання і об'єднання словників. Методи items(), keys(), values(). Комплексні словники. Множина (set, frozenset). Масиви, модуль NumPy. Функціональне програмування.

Тема 24. Робота з файлами

Відкриття та закриття файлу, функція open, метод close(). Режими відкриття файлу. Текстові файли. Файли CSV. Бінарні файли, модуль pickle. Читання з файлу. Запис в файл.

Модуль shelve. Модуль OS і робота з файловою системою.

Тема 25. Робота з рядками

Літерали рядків. Отримання підрядка, функції ord і len, пошук в рядку, перебір рядка. Функції і методи рядків. Форматування рядків, метод format. Байтові рядки.

Тема 26. Класи і об'єкти

Визначення класу, методи та атрибути, властивості, анотації властивостей. Посилання на об'єкт класу, конструктори, метод __init(). Визначення класів в модулях та підключення. Клас object. Строкове представлення об'єкту.

Тема 27. Основні вбудовані модулі

Модулі random, math, locale, decimal. Робота з датами і часом, модуль datetime. Процеси та потоки. Графічний інтерфейс.

Теми лабораторних робіт

Лабораторна робота 1. Знайомство з інтегрованим середовищем розробки програм MS Visual C++.NET

Лабораторна робота 2. Підготовка і розв'язання на ПК задач лінійного характеру на мові C/C++

Лабораторна робота 3. Підготовка і розв'язання на ПК задач з розгалуженням на мові C/C++

Лабораторна робота 4. Підготовка і розв'язання на ПК задач з використанням циклів на мові C/C++

Лабораторна робота 5. Підготовка і розв'язання на ПК задач з використанням функцій і макросів на мові C/C++

Лабораторна робота 6. Підготовка і розв'язання на ПК задач обробки масивів на мові C/C++

Лабораторна робота 7. Підготовка і розв'язання на ПК задач оброблення масивів і рядків в стилі C з використанням покажчиків

Лабораторна робота 8. Підготовка і рішення на ПК задач обробки масивів структур і файлів на мові C/C++

Лабораторна робота 9. Підготовка і рішення на ПК задач обробки динамічних структур даних з використанням STL

Лабораторна робота 10. Підготовка і розв'язання на ПК задач з використанням рядків, файлів, потоків і засобів обробки виключень на мові C/C++

Лабораторна робота 11. Знайомство з інтегрованим середовищем розроблення програм PyCharm. Підготовка і розв'язання на ПК задач лінійного характеру

Лабораторна робота 12. Підготовка і розв'язання на ПК задач з використанням розгалужень і циклів на мові Python

Лабораторна робота 13. Підготовка і розв'язання на ПК задач з використанням функцій мові Python

Лабораторна робота 14. Підготовка і розв'язання на ПК задач з використанням модулів мові Python

Лабораторна робота 15. Підготовка і розв'язання на ПК задач оброблення масивів з використанням засобів обробки виключень мові Python

Лабораторна робота 16. Підготовка і розв'язання на ПК задач з використанням словників і засобів обробки виключень мові Python

Лабораторна робота 17. Підготовка і розв'язання на ПК задач оброблення рядків з використанням файлів мові Python

Лабораторна робота 18. Підготовка і розв'язання на ПК задач оброблення структурованих даних мові Python

4. Порядок оцінювання результатів навчання

Система оцінювання сформованих компетентностей 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, лабораторних занять і оцінюється сумою набраних балів (максимальна сума – 60 балів; мінімальна сума, що дозволяє студенту складати 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 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 і 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 (модульного) екзамену. Екзаменаційний білет охоплює програму дисципліни і передбачає визначення рівня знань та ступеня опанування студентами компетентностей отриманих після проходження відповідного модуля (див. табл.).

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.

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

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. Для розв'язання кожної задачі потрібно розробити програму (проект) на алгоритмічній мові C/C++ або Python. Підсумкова оцінка за екзамен є сумою оцінок за кожне завдання.

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 проставляється у відповідній графі екзаменаційної *«Відомості обліку успішності»*.

Підсумкова оцінка з модулю 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.

Підсумкова оцінка з навчальної дисципліни розраховується як середнє з оцінок отриманих за кожний семестр.

Розподіл балів за тижнями

Семестр 1

Теми змістового модуля			Лекційні заняття	Лабораторні заняття	Індивідуальне завдання	Письмова контрольна робота	Усього
Змістовий модуль 1. Програмування на мовах C/C++. Лексичні основи мов C/C++	ТЕМА 1. Етапи розроблення та впровадження програм ТЕМА 2. Архітектура комп'ютерів, принципи Джона фон Неймана ТЕМА 3. Позиційні системи числення	1 тиждень	0.5	3			3.5
	ТЕМА 4. Елементи алгоритмічних мов C/C++: концепція типів даних, імена, значення, покажчики, змінні, константи, операції, вирази	2 тиждень	0.5	3			3.5
	ТЕМА 5. Структурне програмування: послідовність, розгалуження та цикли	3 тиждень 4 тиждень	1	6			6.5
	ТЕМА 6. Передпроцесорна обробка	5 тиждень	0.5	1			1.5
	ТЕМА 7. Процедурно-орієнтоване програмування. Рекурсія	6 тиждень	0.5	1	2		3.5
	ТЕМА 8. Бібліотеки динамічного компонування (DLL) ТЕМА 9. Методології розроблення програм: низхідне та висхідне проектування, модульне програмування	7 тиждень	0.5	1		6	7.5
	ТЕМА 10. Масиви	8 тиждень	0.5	6			6.5
Змістовий модуль 2. Програмування на мовах C/C++. Основи програмування на мовах C/C++	ТЕМА 11. Похідні типи даних. Рядки в стилі C	9 тиждень 10 тиждень	1	3			3.5
	ТЕМА 12. Структури та об'єднання. Динамічні структури даних	11 тиждень 12 тиждень	1	6			6.5
	ТЕМА 13. Введення в систему вводу-виводу C/C++. Файлові структури даних	13 тиждень	0.5	2			2.5
	ТЕМА 14. Шаблони. Стандартна бібліотека шаблонів	14 тиждень	0.5	3			3.5
	ТЕМА 15. Стандартний клас string	15 тиждень	0.5	2			2.5
	ТЕМА 16. Обробка виключень. Особливості стандартів C11, C++11, C++14	16 тиждень	0.5	1		6	7.5
	Іспит						40
Усього			8	38	2	12	100

Семестр 2

Теми змістового модуля			Лекційні заняття	Лабораторні заняття	Індивідуальне завдання	Письмова контрольна робота	Усього
Змістовий модуль 3. Програмування на мові Python. Лексичні основи мови Python	ТЕМА 17. Синтаксис та семантика мови Python концепція типів даних, іме значення, покажчики, змін константи, операції, вираз	1 тиждень 2 тиждень	1	3			4
	ТЕМА 18. Оператори мови Python	3 тиждень	0.5	3			3.5
	ТЕМА 19. Функції в мові Python	4 тиждень 5 тиждень	1	3			4
	ТЕМА 20. Робота з модулями	6 тиждень	0.5	3			3.5
	ТЕМА 21. Правила написання і документування коду на мові Python	7 тиждень	0.5			6	6.5
Змістовий модуль 4. Програмування на мові Python. Основи програмування на мові Python	ТЕМА 22. Обробка виключень в мові Python	8 тиждень	0.5	3			3.5
	ТЕМА 23. Списки, кортежі і словники	9 тиждень 10 тиждень 11 тиждень	1.5	8			9.5
	ТЕМА 24. Робота з файлами	12тиждень	0.5	3			3.5
	ТЕМА 25. Робота з рядками	13 тиждень	0.5	3			3.5
	ТЕМА 26. Класи і об'єкти	14 тиждень	0.5	3	2		5.5
	ТЕМА 27. Основні вбудовані модулі	15 тиждень 16 тиждень	1	6		6	13
Іспит							40
Усього			8	38	2	12	100

Шкала оцінювання: національна та ЄКТС

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

5. Рекомендована література

Основна

1. Вирт Н. Алгоритмы и структуры данных / Н. Вирт ; пер. с англ. – Москва : Мир, 1989. – 360 с. : ил.
2. Кнут Д. Искусство программирования для ЭВМ / Д. Кнут. – Москва : Мир, Т. 1 – 1997
Т. 2 – 1997
Т. 3 – 1998
3. Страуструп Б. Программирование: принципы и практика использования C++ / Б. Страуструп; пер. с англ. – 2-е изд. – Москва : Издательский дом “Вильямс”, 2016. – 1328 с.
4. Деннис М. Ритчи. Язык программирования C / Деннис М. Ритчи, Брайан У. Керниган ; пер. с англ. – Москва : Издательский дом “Вильямс”, 2009. – 304 с.
5. Ишкова Э. А. C++. Начала программирования / Э. А. Ишкова. – Москва : ООО “БИНОМ-Пресс”, 2004. – 368 с. : ил.
6. Хортон А. Visual C++ 2010: полный курс / А. Хортон ; пер. с англ. – Москва : Издательский дом “Вильямс”, 2011. – 1216 с. : ил.
7. Дейтел Х. Как программировать на C++ / Х. Дейтел, П. Дейтел ; пер. с англ. – Москва : ЗАО “Издательство БИНОМ”, 2008. – 1455 с.
8. Аммерааль Л. STL для программистов на C++ / Л. Аммерааль ; пер. с англ. – Москва : ДМК, 1999. – 240 с. : ил.
9. Щупак Ю. А. Эффективная разработка приложений / Ю. А. Щупак. – Санкт-Петербург : Питер, 2007. – 572 с. : ил.
10. Румянцев П. В. Азбука программирования в Win32 API / П. В. Румянцев. – Москва : Горячая линия – Телеком, 2001. – 312 с.
11. Пирогов В. Ю. Программирование на Visual C++.NET / В. Ю. Пирогов. – Санкт-Петербург : БХВ-Петербург, 2003. – 800 с.
12. Лутц М. Изучаем Python / М. Лутц; пер. с англ. – 4-е изд. – Санкт-Петербург : Символ-Плюс, 2011. – 1280 с. : ил.

Додаткова

13. Страуструп Б. Язык программирования C++. Специальное издание / Б. Страуструп ; пер. с англ. – 3-е изд. – Санкт-Петербург; Москва : “Невский диалект” ; Изд-во “БИНОМ”, 2004. – 1104 с. : ил.
14. Подбельский В. В. Язык C++ : учеб. пособ. / В. В. Подбельский. – 4-е изд. – Москва : Финансы и статистика, 1999. – 560 с.
15. Саммерфилд М. Программирование на Python 3. Подробное руководство / М. Саммерфилд. – Санкт-Петербург : Символ-Плюс, 2009. – 608 с. : ил.
16. Павловская Т. А. C/C++. Программирование на языке высокого уровня / Т. А. Павловская. – Санкт-Петербург : Питер, 2006. – 461 с. : ил.
17. Саймон Р. Windows 2000 API. Энциклопедия программиста / Р. Саймон ; пер. с англ. – Киев : ООО “ДиасофтЮП”, 2002. – 1088 с.
18. Марченко А. Л. C++. Бархатный путь / А. Л. Марченко. – Москва : Горячая линия – Телеком, 1999. – 400 с.
19. Бизли Д. Python. Подробный справочник / Д. Бизли Д. ; пер. с англ. – 4-е изд. – Москва : ДМК, 2012. – 864 с.
20. Подбельский В. В. Программирование на языке C / В. В. Подбельский, С. С. Фомин. – 2-е доп. изд. – Москва : Финансы и статистика, 2004. – 600 с.

21. Шеферд Д. Программирование на Microsoft Visual C++.NET / Д. Шеферд ; пер. с англ. – Москва : Издательско-торговый дом “Русская редакция”, 2003. – 928 с. : ил.

Інформаційні ресурси в Internet

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. Python documentation [Электронный ресурс]. – Режим доступа : <https://docs.python.org/doc/>

25. CodeNet – все для программиста [Электронный ресурс]. – Режим доступа : <http://www.codenet.ru/>.

26. REALCODING – для настоящих программистов [Электронный ресурс]. – Режим доступа : <http://www.realcoding.net/>.

27. MSDN-WindowsAPI. [Электронный ресурс]. – Режим доступа : http://www.vsokovikov.narod.ru/New_MSDN_API/index_msdn.htm.

28. Самоучитель Python [Электронный ресурс]. – Режим доступа : <https://pythonworld.ru/samouchitel-python>

Методичне забезпечення

29. Алгоритмізація та програмування [Електронний ресурс] : навч. посіб. / В. М. Федорченко, О. В. Щербаков, Ю. Е. Парфьонов ; кер. проекту В.М. Анохін; дизайн: Д.О. Романова, програмування: О.К. Борисюк, М.С. Войчук; Харківський національний економічний університет ім. С. Кузнеця. - мультимедійне інтерактивне електрон. вид. комбінованого використ.(413 Мб). - Х. : ХНЕУ ім. С. Кузнеця, 2016. - Загол. з титул. екрану. - ISBN 978-966-676-629-1.

30. 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, О. В. Тарасов. – Харків : Вид. ХНЕУ, 2012. – 180 с.

31. 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. Е. Парфьонов. – Харків : Вид. ХНЕУ, 2009. – 180 с.