

“ЗАТВЕРДЖУЮ”
Завідувач кафедри

В.М.Базилевич

“ _____ ” _____ 2021 р.

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

ОСНОВИ ПРОГРАМУВАННЯ

Рівень вищої освіти – перший (бакалаврський)

Спеціальність: 121 – Інженерія програмного забезпечення

Освітня програма: Інженерія програмного забезпечення

Мова навчання: українська

Статус дисципліни: обов'язкова

Форма навчан.	Рік навч.	Сем.	Розподіл годин					За тиждень		ІНДЗ	Контр.
			Всього ауд.	Лек	Лаб.	СРС	Разом	Ауд.	СРС		
Денна	1	1	60	32	28	150	210	3.8	9.4		Е
	1	2	40	20	20	110	150	2.9	7.9	КП	Е
Усього			60	52	48	260	360				

Робоча програма дисципліни Основи програмування
(назва навчальної дисципліни)

для здобувачів вищої освіти галузі знань 12 – Інформаційні технології спеціальності 121 – Інженерія програмного забезпечення.

Розробник робочої навчальної програми:

доцент кафедри інформаційних та комп'ютерних систем НУ «Чернігівська політехніка», к.т.н., доцент _____ (П.Г. Бивойно)
(підпис) (прізвище та ініціали)

викладач кафедри інформаційних технологій та програмної інженерії НУ «Чернігівська політехніка» _____ (В.В. Нехай)
(підпис) (прізвище та ініціали)

Робочу програму схвалено на засіданні кафедри інформаційних і комп'ютерних систем

Протокол від “27” серпня 2021 року № 1

Завідувач кафедри інформаційних і комп'ютерних систем

_____ (В.М. Базилович)
(підпис) (прізвище та ініціали)

УЗГОДЖЕНО:

Гарант ОП Інженерія програмного забезпечення _____ (І.В. Білоус)
(підпис) (прізвище та ініціали)

Abstract

ESIEIT / SE - Fundamentals of programming (OK16)

2021/2022 Sem. 1

2021/2022 Sem. 2

The object of study of the course is theoretical and practical foundations of modern technology high-level programming for constructing algorithms and data structures used in solving applied problems in various fields of science and technology.

The amount of discipline "Fundamentals of programming" is 12 credits and has two part.

The first part study curriculum of the discipline amount is 7 credits and consists of four content modules:

- the first content module "Basic elements of C" covers issues such as the history of computer technology, programming languages and classification criteria of choice of programming language. Next, we consider the specific issue of language C. This is data types, expressions and operations, functions and operators C.

- the second, short, content module is devoted to the study of pointers and their use for the definition of various types and data structures.

- the third content module addresses the creation and organization of software processing simple data structures – array, strings and structures.

- in the fourth, short, content module we study the use of files for data storage and processing

The two part study curriculum of the discipline amounts is 5 credits and consists of five content modules. The tasks solved during the study of discipline:

- an overview of the conceptual foundations of the daily programs;

- knowledge of the conceptual bases and abstract types in the C ++ code;

- the implementation of common dynamic data structures and recursive algorithms;

- assimilation of the questions of construction and processing of dynamic data structures, in particular lists and multidisciplines, stacks, queues, binary and strongly branched trees and hash tables;

- acquiring practical skills for using dynamic data structures to solve practical programming tasks on C ++.

Contents: history of computer, programming languages, C data types, expressions, operations, functions, operators of C, pointers, arrays, strings, structures, files, dynamic data structures, static data structures, linked list, list based on array of pointers, multy-lists, stack, queue, deck, binary tree, B-tree, hash table, recursion.

1 Опис навчальної дисципліни

Найменування показників	Галузь знань, напрям підготовки, освітньо-кваліфікаційний рівень	Характеристика навчальної дисципліни	
		денна форма навчання	
Кількість кредитів – 12	Галузь знань 12 – Інформаційні технології	Нормативна	
Модулів – 2	Спеціальність: 121 - Інженерія програмного забезпечення	Рік підготовки:	
Змістових модулів – 4		1-й	1-й
Індивідуальні науково-дослідні завдання – курсовий проект		Семестри	
Загальна кількість годин – 360		1-й	2-й
		Лекції	
Тижневих годин: аудиторних – 1 сем.-3.8; 2 сем. -2.9; самостійної та індивідуальної роботи студента – 1 сем.- 9.4; 2 сем. –7.9.	Освітньо-кваліфікаційний рівень: бакалавр	32 год.	20 год.
		Практичні, семінарські	
		Лабораторні	
		28 год.	20 год.
		Самостійна робота	
		150 год.	110 год.
		Індивідуальні завдання:	
			КП.
		Вид контролю:	
Екзамен	Екзамен		

Примітка.

Співвідношення кількості годин аудиторних занять до самостійної і індивідуальної роботи становить:

- у 1-ому семестрі 1:2,5;
- у 2-ому семестрі 1:2,75.

Передумовою для вивчення дисципліни є наявність базових знань шкільного курсу «Інформатика» та навичок роботи з персональним комп'ютером.

Набуті під час вивчення дисципліни знання є базовими для вивчення таких дисциплін, як “ОК21-Об’єктно-орієнтоване програмування”, “ОК24-Java і C# технології прикладного програмування”, “ОК30-Якість програмного забезпечення

та тестування", Набуті знання та вміння застосовуються також у виконанні курсових проектів в подальших семестрах та випускної бакалаврської роботи.

Обов'язковою умовою викладання дисципліни є проведення лабораторного практикуму із застосуванням персональних комп'ютерів та спеціалізованого інтегрованого середовища розробки програм, наприклад, Qt Creator.

2 Мета навчальної дисципліни

Метою викладання навчальної дисципліни "*Основи програмування*" є розвиток загальних та фахових компетентностей бакалавра в галузі знань *12 – Інформаційні технології*.

Зокрема, це такі загальні компетентності, як:

- ЗК1. Здатність до абстрактного мислення, аналізу та синтезу;
- ЗК2. Здатність застосовувати знання у практичних ситуаціях;
- ЗК3. Здатність спілкуватися державною мовою як усно, так і письмово;
- ЗК5. Здатність вчитися і оволодівати сучасними знаннями;
- ЗК6. Здатність до пошуку, оброблення та аналізу інформації з різних джерел;
- ЗК13. Здатність працювати в міжнародному контексті.

Також такі фахові компетенції:

- ФК15. Здатність ідентифікувати, класифікувати та формулювати вимоги до програмного забезпечення;
- ФК16. Здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування;
- ФК21. Володіння знаннями про інформаційні моделі даних, здатність створювати програмне забезпечення для зберігання, видобування та опрацювання даних;
- ФК24. Здатність накопичувати, обробляти та систематизувати професійні знання щодо створення і супроводження програмного забезпечення та визнання важливості навчання протягом всього життя;
- ФК27. Здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення;
- ФК28. Здатність до алгоритмічного та логічного мислення.

3 Очікувані результати навчання з дисципліни

В результаті вивчення дисципліни у студентів має бути сформована чітка система уявлень про цілісний комплекс проблем, що виникають та мають бути вирішені у процесі підготовки програмного проекту для комп'ютера на алгоритмічній мові високого рівня при розв'язанні найбільш поширених практичних задач. Зокрема:

- мати уявлення про системи і середовищами програмування, класифікацією мов програмування та критеріями вибору мови програмування;
- володіти базовими засадами мов програмування високого рівня на

прикладі мови С;

– знати базові алгоритмами обробки простих структур даних та отримання практичних навичок створення таких структур та програмної реалізації алгоритмів їх обробки;

– практично засвоїти навички створення простих структур даних та програмної реалізації алгоритмів їх обробки;

– засвоїти основи роботи з вказівниками;

– познайомитися з поняттям файл.

– ознайомитися з найбільш поширеними динамічними структурами даних та рекурсивними алгоритмами;

– засвоїти питання побудови та обробки динамічних структур даних, зокрема списків і мультисписків, стеків, черг, бінарних і сильнорозгалужених дерев та хеш-таблиць;

– набути практичних навичок використання динамічних структур даних для вирішення завдань практичного програмування.

У результаті вивчення навчальної дисципліни студент повинен

Знати:

1) методи і технології програмування;

2) базові і абстрактні типи даних;

3) методи конструювання алгоритмів;

4) побудову процедур і функцій;

5) способи доступу до даних;

6) можливості вводу та виводу даних;

7) методи та засоби управління процесом обробки даних;

8) абстракції основних структур даних і прості методи їх обробки;

9) найбільш поширені динамічні структури даних: списки та мультисписки; черги; стеки; бінарні та сильнорозгалужені дерева, хеш-таблиці;

10) організацію обробки даних в динамічній пам'яті – "кучі";

11) організацію обробки даних на зовнішніх носіях – файлах.

12) рекурсивні процедури та функції;

13) методи обходу та балансування дерев;

14) засоби організації та пошуку даних в хеш-таблицях.

Вміти:

15) розробити схему алгоритму обробки даних;

16) обирати або створювати типи даних виходячи з особливостей, вирішуваних задач;

17) створювати динамічні структури даних різного призначення;

18) реалізовувати алгоритми на мові високого рівня;

19) спроектувати інтерфейс користувача програмного проекту;

20) організувати обробку динамічних структур даних в "кучі" та в файлах;

21) працювати в середовищах програмування.

Навчальна дисципліна “*Основи програмування*” має допомогти сформулювати наступні програмні результати навчання:

– ПР01. Аналізувати, цілеспрямовано шукати і вибирати необхідні для

вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки;

- ПР04. Знати і застосовувати професійні стандарти і інші нормативно-правові документи в галузі інженерії програмного забезпечення;
- ПР05. Знати і застосовувати відповідні математичні поняття, методи доменного, системного аналізу для розробки програмного забезпечення ;
- ПР07. Знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення;
- ПР08. Вміння розробляти людино-машинний інтерфейс;
- ПР12. Застосовувати на практиці ефективні підходи щодо проектування програмного забезпечення;
- ПР13. Знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань;
- ПР15-мотивовано обирати мови програмування та технології розробки для розв'язання завдань створення і супроводження програмного забезпечення;
- ПР18. Знати та вміти застосовувати інформаційні технології обробки, зберігання та передачі даних.
- ПР23. Вміти документувати та презентувати результати розробки програмного забезпечення.
- ПР25. Мати навички виконання певних ролей в ІТ-проектах будь-якої складності.

4 Критерії оцінювання результатів навчання

Відповідно до [«Положення про поточне та підсумкове оцінювання знань здобувачів вищої освіти Національного університету «Чернігівська політехніка»](#), екзаменаційна оцінка має дві складових:

- результат роботи у семестрі – вага 60%
- результат отриманий безпосередньо на іспиті – вага 40%

До іспиту допускаються здобувачі, що виконали усі заплановані на семестр завдання з підсумковою оцінкою не менше 60 балів (за 100 – бальною шкалою оцінювання). Окрім того, відповідно до [«Положення про поточне та підсумкове оцінювання знань здобувачів вищої освіти Національного університету «Чернігівська політехніка»](#), такі здобувачі, за бажанням, можуть отримати оцінки «достатньо» та «задовільно» без іспиту.

Здобувачі, що мають підсумкову оцінку за семестр менше ніж 35 балів (за 100-бальною шкалою оцінювання) потребують повторного вивчення дисципліни.

З тими здобувачами, які до проведення підсумкового семестрового контролю не встигли виконати всі обов'язкові види робіт та мають підсумкову оцінку за семестр від 35 до 59 балів (за 100-бальною шкалою оцінювання), проводяться додаткові індивідуальні заняття, за результатами яких визначається рівень засвоєння матеріалу, та чи необхідне повторне вивчення дисципліни.

Дисципліну можна вважати такою, що засвоєна, якщо здобувач:

знає:

- що таке тип даних, змінна, константа;
- оператори присвоювання;
- правила написання функцій та звертання до них;
- прості типи даних;
- що таке вказівник і операції з вказівниками;
- правила визначення масивів;
- використання вказівників для обробки масивів.
- хоча б один метод сортування масивів;
- основні алгоритми обробки масивів (переміщення по масиву, знаходження суми елементів, знаходження максимального або мінімального елемента, оборот масиву);
 - як визначити та обробляти структури;
 - поняття статичної та динамічної пам'яті;
 - види динамічних структур даних;
 - правила опису типів для типізованих вказівників та списків;
 - правила побудови однозв'язного списку;
 - алгоритми додавання, вилучення елемента з відсортованого однозв'язного зв'язаного списку;
 - поняття стеку та черги;
 - принципи побудови та основні операції зі стеком та чергою;
 - поняття рекурсивної процедури та рекурсивної структури даних;
 - поняття бінарне дерево, впорядковане бінарне дерево, майже повне бінарне дерево, AVL дерево, RB-дерево;
 - правила обходу бінарних дерев;
 - поняття таблиць хешування, функція хешування, поняття колізії, алгоритми вирішення колізій;
 - поняття рехешування та його види;
 - основні операції з реалізації хеш-таблиць;

вміє:

- працювати у середовищі Qt-Creator або іншому середовищі;
- написати і виконати просту програму на С для обчислення по формулі;
- написати функцію по обробці масиву;
- написати функцію по обробці файлу структур;
- знаходити прості помилки у програмах;
- написати програму, яка створює однозв'язний список;
- написати програму, яка створює однозв'язний список в типізованому файлі, виконує пошук елементів, та їх видалення;
- реалізувати стек та чергу на базі списку або масиву;
- написати рекурсивну процедуру додавання чи вилучення елемента зі списку;

- виконати обхід бінарного дерева у прямому, симетричному, зворотному та горизонтальному напрямку;
- написати функцію хешування для хеш-таблиці.

5 Засоби діагностики результатів навчання

Засобами оцінювання результатів навчання з дисципліни є:

- співбесіда із здобувачем по результатах виконання лабораторних робіт;
- виконання тестових завдань з програмування під час лабораторних робіт;
- курсовий проект;
- іспит.

6 Програма навчальної дисципліни

Змістовий модуль 1. Базові засади програмування

Тема 1. Вступ

Предмет і мета вивчення дисципліни “*Основи програмування*”. Основні задачі, що вирішуються під час вивчення дисципліни. Структура навчального курсу. Навчально-методична література з дисципліни.

Історія розвитку обчислювальної техніки. Тенденції розвитку і перспективи Досягнення вітчизняних вчених та інженерів у цій галузі.

Класифікація мов програмування. Вибір мови програмування.

Системи і середовища програмування. Класифікація програмного забезпечення та вимоги до системних програм.

Сучасні інформаційно-довідникові ресурси, що необхідні для вирішення професійних завдань та способи їх пошуку.

Англійська мова та англійськомовні інформаційно-довідникові ресурси як запорука здатності працювати в міжнародному контексті.

Тема 2. Базові елементи мови C

Поняття про програму на мові C. Основні риси мови програмування C. Алфавіт мови. Лексеми. Структура програми на C. Директиви препроцесора. Поняття про бібліотечні функції та заголовні файли. Етапи виконання програми.

Використання об'єктів C++ cout, cin для введення-виведення даних. Створення простішого діалогу з консоллю.

Тема 3. Типи даних

Класифікація типів даних. Цілочислові типи. Дійсні типи. Описи змінних. Переліки. Бібліотека математичних функцій.

Тема 4. Вирази та операції

Арифметичні та порозрядні операції. Операції порівняння та логічні операції. Операції присвоєння, комбіновані присвоєння. Умовна операція (тернарний оператор). Операція розміру sizeof. Порядок виконання операцій. Узгодження типів операндів у виразах.

Тема 5. Функції

Функція як ефективний засіб проектування програмного забезпечення.

Доменний аналіз як засіб накопичення, систематизації і збереження минулого досвіду для повторного використання.

Структура функцій. Правила написання функцій. Виклик функції. Прототип функції. Взаємодія фактичних і формальних параметрів. Рекурсивні функції. Области оголошення та доступу до імен. Макроси з параметрами.

Тема 6. Оператори мови C

Поняття про схеми алгоритмів та їх різновиди. ДСТУ ISO 5807:2016.

Оператори-вирази: присвоєння, виклик функції, порожній оператор. Умовні оператори: if, switch. Оператори циклу: for, while, do-while.

Оператори переходу: goto, break, continue, return.

Ітераційні, рекурентні, рекурсивні алгоритми. Випадкові числа.

Змістовий модуль 2. Вказівники**Тема 7. Базові поняття про вказівники**

Вказівники як ефективний засіб розробки програмного забезпечення.

Оголошення вказівників. Звертання до даних через вказівники. Адресна арифметика. Вказівники void. Типізація вказівників.

Змістовий модуль 3. Створення та обробка простих структур даних**Тема 8. Одновимірні масиви**

Оголошення та ініціалізація масивів. Звертання до елементів масиву через індекси. Обробки масивів з використанням вказівників Реалізація простих алгоритмів обробки масивів.

Сортування масивів. Реалізація алгоритмів роботи з упорядкованими масивами.

Тема 9. Рядки символів

Оголошення та ініціалізація рядків символів. Звертання до елементів рядків через індекси та вказівники. Реалізація алгоритмів обробки рядків символів.

Масиви покажчиків на рядки символів.

Тема 10. Структури

Оголошення та ініціалізація структур. Операція присвоєння для структур . Звертання до полів структури. Декларація іменування типу typedef.

Покажчики на структури.

Змістовий модуль 4. Файли**Тема 11. Використання файлів для збереження та обробки даних**

Типи файлів у мові C. Відкриття та закриття файлу. Використання послідовного доступу до даних у файлі. Організація прямого доступу до даних.

Тема 12. Закінчення

Ретроспективний огляд тем курсу.

Ролі учасників розробки реальних ІТ-проектів та значення навичок програмування для успішного виконання цих ролей.

Короткий огляд наступних курсів, пов'язаних з програмуванням.

Змістовий модуль 5. Вказівники та зв'язані списки, мультисписки.

Тема 13. Структури та типи даних. Абстрактні типи даних

Поняття базових структур і типів даних в мові C++. Можливості мов програмування по побудові абстрактних структур і типів даних. Проблема моделювання динамічних даних.

Визначення поняття структури даних. Приклади структур даних. Методи по обробці структур даних. Динамічні структури даних. Історія появи динамічних структур даних та їх місце в сучасному програмуванні.

Визначення поняття типу даних. Концепція базових та абстрактних типів даних.

Тема 14. Спискові структури та їх різновиди

Лінійні списки. Списки з заголовками. Циклічні списки. Циклічні списки з заголовками.

Способи реалізації лінійних списків: вузлове представлення, представлення масивами вказівників, представлення динамічними масивами.

Тема 15. Однозв'язні списки та їх реалізація

Операції з лінійними списками в залежності від способу представлення. Вставка, вилучення, пошук і заміна елемента списку та інше. Сортування списків.

Тема 16. Двонапрямкові списки та їх реалізація

Двонапрямкові списки. Їх реалізація та операції з ними.

Тема 17. Мультисписки їх види та реалізація

Мультисписки. Призначення та використання мультисписків. Їх реалізація та операції з ними.

Тема 18. Файловий ввід-вивід

Основи файлової системи.

Основні функції роботи з файлами fopen(), fseek(), feof(), fread(), fwrite(), fclose(). Використання getc(), putc(), fputs() і fgets().

Ввід/вивід при прямому доступі до файлу.

Організація спискових структур у файлі. Їх реалізація та операції з ними.

Змістовий модуль 6. Спискові структури на базі масивів вказівників

Тема 19. Однозв'язні списки на базі масивів вказівників.

Безтипові вказівники та засоби робот з ними. Побудова динамічної спискової структури на базі масиву вказівників та її обробка.

Реалізація спискової структури на основі масиву вказівників у безтиповому файлі.

Тема 20. Мультиспискові структури на основі списків на базі масивів вказівників.

Мультиспискові структури на базі масивів вказівників у динамічній пам'яті.

Мультиспискові структури на базі масивів вказівників у безтипових файлах.

Змістовий модуль 7. Абстрактні типи даних, рекурсія.

Тема 21. Реалізація абстрактних типів даних (стеки, черги, деки)

Масиви та проблема їх реалізації в пам'яті комп'ютера. Реалізація одно - дво - та N-мірних масивів. Стеки та їх методи. Реалізація стеків з допомогою масивів і списків.

Черги та їх методи. Реалізація черг з допомогою масивів і списків.

Тема 22. Використання стеків та черг

Стеки - як приклад правильного вибору типу даних для ефективного вирішення складних задач. Реалізація та використання ПОЛІЗ з допомогою стеків.

Область використання черг.

Тема 23. Рекурсія. Рекурсивні структури даних та їх обробка

Рекурсивні об'єкти та рекурсивні процедури. Застосування рекурсивних процедур до рекурсивних об'єктів. Рекурсивні процедури обробки списків. Рекурсивний пошук в лабіринті. Стеки та рекурсивні процедури. Умови застосування рекурсії.

Тема 24. Шаблони

Класи та об'єкти. Інкапсуляція, наслідування поліморфізм. Конструктори та деструктори.

Стандартна бібліотека шаблонів STL.

Реалізація абстрактних типів даних засобами шаблонів STL.

Змістовий модуль 8. Бінарні та сильноразгалуджені дерева**Тема 25. Бінарні дерева та їх представлення**

Види бінарних дерев. Представлення бінарних дерев за допомогою масивів та спискових структур.

Тема 26. Збалансовані бінарні дерева

AVL-дерева та Збалансовані RB – дерева. Методи їх балансування. Вилучення та включення елемента у збалансовані дерева.

Тема 27. Сильноразгалуджені дерева

Представлення, проходження та використання сильноразгалуджених дерев.

Змістовий модуль 9. Хешування.**Тема 28. Принципи побудови хеш-таблиць, та їх призначення.**

Основні положення. Концепція хешування. Типи алгоритмів хешування – відкрите та закрите хешування.

Спискові алгоритми відкритого хешування.

Алгоритми закритого хешування. Додавання елементів в таблицю. Метод лінійного хешування. Пошук та вилучення елементів з таблиці. Вирішення колізій. Первинна кластеризація. Алгоритми рехешування.

Блочні алгоритми відкритого хешування. Базова структура блочного хешування. Зв'язування блоків. Додавання та вилучення елементів в блочному алгоритмі.

Тема 29. Функції хешування.

Види функцій хешування. Вибір функції хешування. Принципи та вимоги створення функцій хешування.

Функції рехешування.

7 Структура навчальної дисципліни

Назви змістовних модулів і тем	Кількість годин			
	У тому числі			
	Лекції	Лабораторні	Самостійна робота	Всього
1	2	3	4	5
СЕМЕСТР 1				
Змістовий модуль 1. Базові засади програмування.	16	14	94	124
1. Вступ	1	1	2	4
2. Базові елементи мови C	1	1	2	4
3. Типи даних	2	1	4	7
4. Вирази та операції	2	1	2	5
5. Функції	2	2	8	12
6. Оператори мови C	8	8	76	92
Змістовий модуль 2. Вказівники.	2	2	8	12
7. Базові поняття про вказівники	2	2	8	12
Змістовий модуль 3. Створення та обробка простих структур даних.	10	10	40	60
8. Одновимірні масиви	6	6	24	36
9. Рядки символів	2	2	8	12
10. Структури	2	2	8	12
Змістовий модуль 4. Файли.	4	2	8	14
11. Використання файлів для збереження та обробки даних	2	2	8	12
12. Підсумки	2			2
Всього за 1 семестр	32	28	150	210
СЕМЕСТР 2				
Змістовий модуль 5. Вказівники та зв'язані списки, мультисписки.	7	8	15	30
13. Структури та типи даних. Абстрактні типи даних.	1		1	2
14. Спискові структури та їх різновиди.	1		2	3
15. Однозв'язні списки та їх реалізація.	1	2	2	5
16. Двонапрямкові списки та їх реалізація.	1	2	2	5
17. Мультисписки їх види та реалізація.	1	2	4	7
18. Файловий ввід-вивід.	2	2	4	8

Змістовий модуль 6. Спискові структури на базі масивів вказівників.	2	2	26	30
19. Однозв'язні списки на базі масивів вказівників.	1		12	13
20. Мультиспискові структури на основі списків на базі масивів вказівників.	1	2	14	17
Змістовий модуль 7. Абстрактні типи даних, рекурсія.	6	6	18	30
21. Реалізація абстрактних типів даних (стеки, черги, деки).	2	2	4	8
22. Використання стеків та черг.	1		4	5
23. Рекурсія. Рекурсивні структури даних та їх обробка.	2	2	5	9
24. Шаблони.	1	2	5	8
Змістовий модуль 8. Бінарні та сильноразгалуджені дерева.	3	2	25	30
25. Бінарні дерева та їх представлення.	1	2	8	11
26. Збалансовані бінарні дерева.	1		9	10
27. Сильноразгалуджені дерева	1		8	9
Змістовий модуль 9. Хешування.	2	2	26	30
28. Принципи побудови хеш-таблиць, та їх призначення.	1		13	14
29. Функції хешування.	1	2	13	16
Всього за 2 семестр	20	20	110	150

8 Теми лабораторних занять

№ з/п	Назва теми	Кількість годин
СЕМЕСТР 1		28
1	Правила техніки безпеки та протипожевної безпеки в лабораторії. Знайомство з інтегрованим середовищем розробки (ICP)“QT-creator”	2
2	Типи даних	2
3	Функції та макроси з параметрами	2
4	Логічний тип даних і розгалуження у програмах	2
5	Побудова циклів з операторами „while” і „do...while”	4
6	Обробка даних за допомогою циклу for	2
7	Використання покажчиків	2
8	Рядки символів	2
9	Одновимірні масиви	2
10	Сортування масивів	2
11	Обробка впорядкованих масивів	2
12	Робота із структурами	2
13	Файли	2
СЕМЕСТР 2		20
1	Алгоритми обробки лінійного однозв'язного списку.	2

2	Алгоритми обробки лінійного двозв'язного списку.	2
3	Реалізація мультиспискової структури ієрархічного типу на базі лінійних зв'язних списків.	2
4	Реалізація лінійного списку в двійковому файлі.	2
5	Реалізація мультисписку ієрархічного типу на базі масивів вказівників на невизначений тип.	2
6	Стеки, черги та деки – реалізація на базі списків і масивів.	2
7	Реалізація рекурсивних функцій для обробки списків	2
8	Знайомство з бібліотекою шаблонів STL мови C++	2
9	Алгоритми хещування. Розв'язання колізій.	2
10	Реалізація впорядкованого бінарного дерева пошуку як спискової структури.	2
		48

9 Самостійна робота

№ з/п	Назва теми	Кількість годин
1	Поточне опрацювання теоретичного матеріалу	104
2	Підготовка до лабораторних робіт	96
4	Виконання курсового проекту	40
5	Підготовка до іспитів	20
Разом		260

10 Індивідуальні завдання

Робочим планом передбачено виконання індивідуальних завдань у вигляді завдань для самостійної роботи, де здобувачі, використовуючи набуті теоретичні знання, розробляють невеликі програми обробки різних структур даних. Варіанти завдань містяться у методичних вказівках до лабораторних робіт [14.1,14.2].

У другому семестрі передбачено виконання індивідуальних завдань у вигляді курсового проекту. Вимоги до курсового проекту містяться у методичних вказівках до виконання проекту [14.3]. Виконавши роботу, здобувач має представити звіт та презентувати роботу.

Робочим планом передбачено виконання індивідуальних завдань у вигляді курсового проекту, коли студенти, базуючись на теоретичних знаннях, набутих в курсі «Основи програмування», на конкретних прикладах самостійно розробляють програми обробки різних структур даних.

Метою виконання курсового проекту є закріплення здобувачами отриманого на лекціях теоретичного матеріалу, вдосконалення практичних навичок здобутих на лабораторних роботах та перевірка вміння студента самостійно створювати складні програми або виконувати групову програмну

розробку. Варіанти завдань до курсових проектів містяться в відповідних методичних вказівках.

При виконанні курсового проекту студент має розробити мультиспискову структуру, що підтримує структуру заданої предметної області та забезпечує ініціалізацію структури, пошук, додавання, вилучення та заміну елемента структури на будь-якому рівні мультисписку з підтримкою цілісності мультисписку; відображення структури мультисписку за допомогою компонент treeview та listview, збереження мультисписку в файлі та відтворення його з файлу.

Форми контролю та оцінювання виконання курсового проекту наведені в таблиці.

Вид роботи	Форма контролю	Кількість балів	
Програмна частина	1. Відповідність умовам завдання	0...	20
	2. Відповідність вимогам стандартів	0...	10
Пояснювальна записка	1. Обґрунтованість технічних рішень	0...	20
	2. Посилання на першоджерела	0...	5
	3. Відповідність оформлення вимогам	0...	15
	4. Своєчасність здачі	0...	10
Презентація курсового проекту	Самостійність виконання (відповіді на запитання)	0...	20
Разом		0...	100

11 Методи контролю

Оцінювання знань здобувачів здійснюється відповідно до [«Положення про поточне та підсумкове оцінювання знань здобувачів вищої освіти Національного університету «Чернігівська політехніка»](#).

З дисципліни здобувач може набрати до 60% підсумкової оцінки за виконання всіх видів робіт, що виконуються протягом семестру і до 40% підсумкової оцінки – на іспиті.

Поточний контроль проводиться шляхом спілкування із здобувачами під час співбесід по темах лабораторних занять та по результатам виконання тестових завдань з програмування [14.1,14.2].

Семестровий контроль за результатами вивчення дисципліни проводиться в останній атестаційний тиждень семестру. Оцінка виставляється у відповідну графу екзаменаційної відомості.

На іспиті здобувач готує відповіді на три питання білету і після цього спілкується з викладачем. Кожне питання оцінюється за 12-бальною шкільною системою. Ще до чотирьох балів здобувач може отримати, відповідаючи на додаткові питання. Таким чином студент може отримати до 40 балів.

Засобом оцінювання результатів навчання з дисципліни у 2-му семестрі є курсовий проект. Оцінка за проект виставляється за 100 бальною шкалою і

враховує:

- оцінки керівника проекту, що враховує своєчасність завершення проекту, його складність, якість коду та оформлення;
- оцінки комісії за презентацію результатів виконаних завдань та досліджень. Здобувач має зробити коротке повідомлення про тему роботи і шляхи вирішення завдань, які були поставлені у технічному завданні. Окрім того здобувач має дати відповіді на практичні і теоретичні питання.

12 Розподіл балів, які отримують студенти

12.1 Поточний контроль за модулями

Поточний контроль за результатами лабораторних робіт

		Кількість балів	
Усього		0...	100
	Підготовленість до лабораторної роботи.	0...	10
	Самостійність виконання лабораторної роботи.	0...	25
	Своєчасність виконання лабораторної роботи.	0...	15
	Повнота і якість оформлення звіту.	0...	6
	Теоретичне питання	0...	22
	Практичне завдання	0...	22

Для захисту лабораторної роботи здобувач повинен відповісти на всі контрольні запитання з методичних вказівок та на запитання за вибором викладача з лекційного курсу за темою лабораторної роботи та виконати практичне завдання. За кожен лабораторну роботу здобувач отримує певну кількість балів згідно наведеної вище таблиці. При цьому враховується якість оформлення звіту та повнота відповідей на запитання при захисті лабораторної роботи.

У кінці навчального семестру обчислюється сума балів, яку здобувач отримує за виконання лабораторних робіт. Бали за семестр обчислюються за результатами лабораторних робіт та тестових контрольних, як середнє арифметичне усіх результатів за 100-бальною шкалою. Отримане середнє значення множимо на 0,6, що і буде результатом поточного контролю.

Під час семестрового контролю до визначеної суми додається оцінка за відповідь на завдання іспиту відповідно 12.2, що відображається в екзаменаційній відомості. Екзаменаційна оцінка виставляється відповідно до шкали оцінювання, наведеної в таблиці 12.3.

12.2 Підсумковий контроль

Іспит		Кількість балів
1	Теоретичне питання із практичним завданням по темі питання	0... 12
2	Теоретичне питання із практичним завданням по темі питання	0... 12

Іспит		Кількість балів
3	Практичне завдання із практичним завданням по темі питання	0... 12
4	Додаткові питання	0... 4
Оцінка за іспит		0... 40

12.3 Шкала оцінювання: національна та ECTS

Сума балів за всі види навчальної діяльності	Оцінка ECTS	Оцінка за національною шкалою	
		для іспиту, курсової роботи	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 з можливістю повторного складання	не зараховано з можливістю повторного складання
0-34	F	незадовільно з обов'язковим повторним вивченням дисципліни	не зараховано з обов'язковим повторним вивченням дисципліни

13 Інструменти, обладнання та програмне забезпечення, використання яких передбачає навчальна дисципліна

Лекційний матеріал подається у вигляді презентацій за допомогою медіа-проектора. Під час лекцій аналізуються проблемні ситуації, підтримується зворотний зв'язок з аудиторією шляхом формулювання запитань і стислих відповідей з обох сторін.

Особливістю виконання лабораторних робіт є застосування програмного забезпечення, а саме – IDE Qt Creator.

14 Методичне забезпечення

1. Основи програмування. Методичні вказівки до лабораторного практикуму та самостійної роботи з дисципліни «Основи програмування» для студентів спеціальностей 121 – «Інженерія програмного забезпечення» 123 – «Комп'ютерна інженерія». /Укл.: Бивойно П.Г.– Чернігів: ЧНТУ, 2019.– 143 с.
2. Основи програмування. Методичні вказівки до лабораторного практикуму та самостійної роботи з дисципліни «Основи програмування. Частина 2» для студентів спеціальностей 121 – «Інженерія програмного забезпечення». /Укл.: Гребенник А.Г. – Чернігів: ЧНТУ, 2019. – 135с.

3. Реалізація структур даних мовою програмування С : метод. вказівки до викон. курс. проекту з дисц. «Основи програмування» для студ. спец. 121 “Інженерія програмного забезпечення”, 123 “Комп’ютерна інженерія” / уклад.: П. Г. Бивойно ; Каф. ІКС. – Чернігів : РВВ ЧНТУ, 2020. – 51 с. – Комп’ютерний файл. Режим доступу: <http://ir.stu.cn.ua/handle/123456789/19228>
4. Програмування. Конспект лекцій з дисципліни «Програмування» для студентів напряму підготовки 6.050103 – „Програмна інженерія”. /Укл.: Бивойно П.Г. - Чернігів: ЧНТУ, 2014. - 139 с.

15 Рекомендована література

Базова

1. Шпак З.Я. Програмування мовою С. – Львів: Оріяна-Нова, 2012. – 432 с.
2. Татарчук Д. Д., Діденко Ю. В. Програмування мовами С та С++: навч. посіб. / Д.Д. Татарчук, Ю.В. Діденко. – К.: , 2012. – 112 с.
3. Вінник В.Ю. Алгоритмічні мови та основи програмування мовою С. – Житомир: ЖДТУ, 2007.- 328 с.
4. Васильєв, О. Програмування на С++ в прикладах і задачах : навч. посіб. / О. Васильєв. – Київ : Ліра-К, 2017. – 381 с.
5. Козак, Л. І. Основи програмування : навч. посіб. : рек. МОН України / Л. І. Козак, І. В. Костюк, С. Л. Стасевич. – Львів : Новий Світ - 2000, 2019. – 325 с.

Допоміжна

1. Kernighan B. W., Ritchie D. M. C Programming Language / Dennis M. Ritchie, Brian W. Kernighan, – 2nd ed. – Prentice-hall, inc., 1988. – 263 p.
2. Шилдт Герберт. Полный справочник по С++. Пер. с англ. – М: Вильямс, 2004. 783 с.
3. Алгоритмы и структуры данных. Новая версия для Оберона + CD / Пер.с англ. Ткачев Ф. В. – М.: ДМК Пресс, 2011. – 272 с.

16 Інформаційні ресурси

1. C Programming Tutorial. – Режим доступу: <https://www.guru99.com/c-programming-tutorial.html>
2. Learn C and C++ Programming. – Режим доступу: <https://www.cprogramming.com/>
3. 9 безкоштовних книг з мови Сі для початківців і не лише...(англійською мовою) – Режим доступу: <https://echo.lviv.ua/dev/7366>
4. Moodle. Основи програмування (Частина 1) – Режим доступу: <https://eln.stu.cn.ua/login/index.php>
5. Moodle. Основи програмування (Частина 2) – Режим доступу: <https://eln.stu.cn.ua/course/view.php?id=3490>

Додаток А

Інформація про забезпечення реалізації та контролю компетенцій в курсі
«Основи програмування»

Компетенція	Забезпечення реалізації	Контроль
Загальні компетенції		
ЗК1. Здатність до абстрактного мислення, аналізу та синтезу	Завдання до самостійної роботи в лабораторних роботах та курсовий проєкт	Екзаменаційні завдання. Результат виконання курсового проєкту.
ЗК2. Здатність застосовувати знання у практичних ситуаціях	Завдання до самостійної роботи в лабораторних роботах та курсовий проєкт	Екзаменаційні завдання. Результат виконання курсового проєкту.
ЗК3. Здатність спілкуватися державною мовою як усно, так і письмово	Методичні матеріали та викладання курсу українською мовою.	Спілкування зі здобувачами українською мовою.
ЗК5. Здатність вчитися і оволодівати сучасними знаннями	Виконання усіх складових навчальної програми курсу	Прийом лабораторних робіт, захист курсового проєкту, іспит.
ЗК6. Здатність до пошуку, оброблення та аналізу інформації з різних джерел	Тема 1. Вступ. Абзаци 1, 5. Виконання лабораторних робіт та курсового проєкту	Екзаменаційне питання №2. Захист курсового проєкту.
ЗК13. Здатність працювати в міжнародному контексті.	Тема 1. Вступ. Абзац 6. Під час виконання лабораторних робіт	Екзаменаційне питання №56. Захист лабораторних робіт.
Фахові компетенції		
ФК15. Здатність ідентифікувати, класифікувати та формулювати вимоги до програмного забезпечення (частково)	Тема 1. Вступ. Абзац 4. Конспект. Презентація лекції.	Екзаменаційне питання № 3. Захист курсового проєкту.
ФК16. Здатність брати участь у	Тема 6, абзац1.	Захист

проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування(частково)	Лабораторні роботи 4,5,6 Виконання курсового проекту.	лабораторних робіт Захист курсового проекту. Екзаменаційне питання 58
ФК21. Володіння знаннями про інформаційні моделі даних, здатність створювати програмне забезпечення для зберігання, видобування та опрацювання даних(частково).	Тема 8, Масиви. Тема 9, Рядки символів. Тема 10, Структури. Тема 11, Використання файлів для збереження та обробки даних. Лабораторні роботи 8-13.	Екзаменаційні питання Екзаменаційні питання 20, 24, 28, 30, 35-44. Захист курсового проекту.
ФК24. Здатність накопичувати, обробляти та систематизувати професійні знання щодо створення і супроводження програмного забезпечення та визнання важливості навчання протягом всього життя(частково)	Тема 1, Вступ, абзац 2. Конспект. Презентація лекції.	Екзаменаційне питання № 4.
ФК27. Здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення(частково)	Тема 1, Вступ, абзац 4. Конспект. Презентація лекції.	Екзаменаційне питання № 5
ФК28. Здатність до алгоритмічного та логічного мислення.	Впродовж всього курсу, зокрема тема 6. Лабораторні роботи 5, 6, 8-10.	Екзаменаційні питання 24, 28, 30,36-38,40,42,43. Захист лабораторних робіт
Програмні результати навчання:		
ПР01. Аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки	Тема 1. Вступ. Абзаци 1, 5. Конспект. Презентація лекції. Під час виконання лабораторних робіт і курсового проекту	Екзаменаційне питання №2. Захист курсового проекту.
ПР04. Знати і застосовувати	Тема 6. Абзац 1.	Прийом звітів з

професійні стандарти і інші нормативно-правові документи в галузі інженерії програмного забезпечення.	Методичні вказівки до лабораторних робіт та курсового проєкту. Вимоги до оформлення звітів.	лабораторних робіт та курсового проєкту
ПР05. Знати і застосовувати відповідні математичні поняття, методи доменного, системного аналізу для розробки програмного забезпечення (частково)	Тема 5, Функції, абзац 2	Екзаменаційне питання №57
ПР07. Знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення (початкові навички)	Виконання лабораторних робіт та курсового проєкту	Захист курсового проєкту
ПР08. Вміння розробляти людино-машинний інтерфейс	Методичні вказівки до виконання курсового проєкту. Виконання лабораторних робіт та курсового проєкту	Захист лабораторних робіт і курсового проєкту
ПР12. Застосовувати на практиці ефективні підходи щодо проектування програмного забезпечення(частково)	Тема 5, Функції, абзац 1. Тема 7. Вказівники, абзац 1 Конспект. Презентація лекції Лабораторні роботи 3-13.	Екзаменаційні питання 18, 48, 49
ПР13. Знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань (частково)	Тема 6. Лабораторні роботи 5, 6, 8-10. Тема 8, Масиви. Тема 9, Рядки символів. Тема 10, Структури. Тема 11, Використання файлів для збереження та обробки даних. Лабораторні роботи 8-13.	Екзаменаційні питання 20, 24, 28, 30, 35-44. Захист лабораторних робіт.

<p>ПР15. Мотивовано обирати мови програмування та технології розробки для розв'язання завдань створення і супроводження програмного забезпечення (частково)</p>	<p>Тема 1, Вступ, абзац 4. Конспект. Презентація лекції.</p>	<p>Екзаменаційне питання № 5</p>
<p>ПР18. Знати та вміти застосовувати інформаційні технології обробки, зберігання та передачі даних (частково).</p>	<p>Тема 6. Лабораторні роботи 5, 6, 8-10. Тема 8, Масиви. Тема 9, Рядки символів. Тема 10, Структури. Тема 11, Використання файлів для збереження та обробки даних. Лабораторні роботи 8-13.</p>	<p>Екзаменаційні питання 20, 24, 28, 30, 35-44. Захист лабораторних робіт.</p>
<p>ПР23. Вміти документувати та презентувати результати розробки програмного забезпечення.</p>	<p>Тема 6. Абзац 1. Підготовка звітів з лабораторних робіт. Підготовка звіту та презентації до захисту курсового проєкту.</p>	<p>Звіти з лабораторних робіт, Захист курсового проєкту</p>
<p>ПР25. Мати уявлення про ролі учасників розробки реальних ІТ-проєктів.</p>	<p>Тема 12. Абзац 2.</p>	<p>Екзаменаційне питання 59</p>