

Міністерство освіти і науки України
Національний університет «Чернігівська політехніка»
Навчально-науковий інститут електронних та інформаційних технологій
Кафедра інформаційних технологій та програмної інженерії

ЗАТВЕРДЖУЮ
Завідувач кафедри

І.В.Білоус

“ _____ ” _____ 2020 року

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

ОСНОВИ ПРОГРАМУВАННЯ (ОК16)

Освітня програма «Інженерія програмного забезпечення»

Рівень вищої освіти – *перший (бакалаврський)*

Спеціальність *121 – 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			100	52	48	260	360				

Чернігів – 2020 рік

Робоча програма дисципліни Основи програмування для здобувачів вищої освіти галузі знань 12 – «Інформаційні технології» спеціальності 121 – «Інженерія програмного забезпечення»

Розробник робочої навчальної програми:

старший викладач кафедри інформаційних технологій та програмної інженерії

_____ (підпис) (А.Г. Гребенник)
(прізвище та ініціали)

Робоча програма схвалено на засіданні кафедри *інформаційних технологій та програмної інженерії*

Протокол від “02” вересня 2020 року №1

Завідувач кафедри *інформаційних технологій та програмної інженерії*

_____ (підпис) (І.В. Білоус)
(прізвище та ініціали)

Abstract

ESIEIT / SE OK16 Fundamentals of Programming

2020/2021 Sem. 1

2020/2021 Sem. 2

Course Description

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 1st 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 - arrays, strings and structures.
- in the fourth, short, content module we study the use of files for data storage and processing

The 2nd 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 – 7	Спеціальність: 121 – Інженерія програмного забезпечення		
Змістових модулів – 9		Семестр	
Індивідуальне науково-дослідне завдання – курсовий проект			
Загальна кількість годин – 360		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.
	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 «Програмування програмного забезпечення», «Моделювання систем-орієнтоване програмування», «Бази даних». 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 закріплення та розвиток загальних та фахових компетентностей бакалавра в галузі знань *12 – Інформаційні технології* із застосування у повсякденній діяльності та розробки програмного забезпечення. 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.
- ЗК31. 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

Навчальна дисципліна «*Основи програмування*» має допомогти сформувати 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.
- ПР05. Знати і застосовувати відповідні математичні поняття, методи доменного, системного і об'єктно-орієнтованого аналізу та математичного моделювання 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. Вміти 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.

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

- мати уявлення про системи та середовища програмування, класифікацію мов програмування та критерії вибору мови програмування;
- володіти базовими засадами мов програмування високого рівня на прикладі мови C;
- знати базові алгоритми 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 сильноразгалуджені 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:

- 1) розробити схему алгоритму обробки даних;
- 2) обирати або створювати типи даних виходячи з особливостей, вирішуваних задач;
- 3) створювати динамічні структури даних різного призначення;
- 4) реалізовувати алгоритми на мові високого рівня;
- 5) спроектувати інтерфейс користувача програмного проекту;
- 6) організовувати обробку динамічних структур даних в "кучі" та в файлах;
- 7) працювати в середовищах програмування.

4 Критерії оцінювання результатів навчання

Відповідно до «Положення...»¹ діючого в університеті, екзаменаційна оцінка має дві складові:

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

До іспиту допускаються студенти, що виконали усі заплановані на семестр завдання з підсумковою оцінкою не менше 60 балів (за 100-бальною шкалою оцінювання). Окрім того, відповідно до «Положення...», такі студенти, за бажанням, можуть отримати оцінки «достатньо» та «задовільно» без іспиту.

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

З тими студентами, 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 студент:

¹ «Положення про поточне та підсумкове оцінювання знань здобувачів вищої освіти Національного університету «Чернігівська політехніка», погодженого вченою радою НУ «Чернігівська політехніка» (протокол № 6 від 31.08.2020 р.) та затвердженого наказом ректора НУ «Чернігівська політехніка» від 31.08.2020 р. №26 <https://www.stu.cn.ua/media/files/pdf/nzop/p-ppocin.pdf>.

знає:

- що таке тип даних, змінна, константа;
- оператори присвоєння;
- 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 C 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 студентом по результатам лабораторних робіт;
 - виконання тестових завдань з програмування під час лабораторних робіт;
 - курсовий проект (2 семестр);
 - іспит.

6 Програма навчальної дисципліни

Змістовий модуль 1. Базові засади програмування

Тема 1. Вступ

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

Історія розвитку обчислювальної техніки. Досягнення вітчизняних вчених та інженерів у цій галузі. Системи і середовища програмування. Класифікація мов програмування. Вибір мови програмування.

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 найпростішого 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

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

Тема 6. Оператори мови C

Поняття про схеми алгоритмів та їх різновиди. 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

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

Змістовий модуль 3. Створення та обробка простих структур даних

Тема 8. 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 9. Одновимірні масиви

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

Тема 10. Впорядковані масиви

Сортування масивів. Реалізація алгоритмів роботи з упорядкованими масивами.

Тема 11. 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 12. Базові поняття про файли

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

Тема 13. Підсумки

Ретроспективний огляд тем курсу. Завдання наступного семестру. Короткий огляд інших курсів, пов'язаних з програмуванням.

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

Тема 14. 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 15. 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 16. 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 17. Двонапрямкові списки та їх реалізація

Двонапрямкові списки. Їх реалізація та операції з ними.

Тема 18. Мультисписки їх види та реалізація

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

Тема 19. 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 20. 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 21. 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 22. 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 23. 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 24. 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 25. 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 сильноразгалужені дерева

Тема 26. Бінарні дерева та їх представлення

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

Тема 27. Збалансовані бінарні дерева

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

Тема 28. Сильноразгалуджені дерева

Представлення, проходження та використання сильноразгалуджених дерев.

Змістовий модуль 9. Хешування.

Тема 29. 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 30. 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. Рядки символів	2	2	8	12
9. Одновимірні масиви	2	2	8	12
10. Впорядковані масиви	4	4	16	24
11. Структури	2	2	8	12
Змістовий модуль 4. Файли.	4	2	8	14
12. Базові поняття про файли	2	2	8	12
13. 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
14. Структури та типи даних. Абстрактні типи даних.	1		1	2
15. Спискові структури та їх різновиди.	1		2	3
16. Однозв'язні списки та їх реалізація.	1	2	2	5
17. Двонапрямкові списки та їх реалізація.	1	2	2	5
18. Мультисписки їх види та реалізація.	1	2	4	7
19. Файловий ввід-вивід.	2	2	4	8
Змістовий модуль 6. Спискові структури на базі масивів вказівників.	2	2	26	30
20. Однозв'язні списки на базі масивів вказівників.	1		12	13
21. Мультиспискові структури на основі списків на базі масивів вказівників.	1	2	14	17
Змістовий модуль 7. Абстрактні типи даних, рекурсія.	6	6	18	30
22. Реалізація абстрактних типів даних (стеки, черги, деки).	2	2	4	8
23. Використання стеків та черг.	1		4	5
24. Рекурсія. Рекурсивні структури даних та їх обробка.	2	2	5	9
25. Шаблони.	1	2	5	8
Змістовий модуль 8. Бінарні та сильноразгалуджені дерева.	3	2	25	30
26. Бінарні дерева та їх представлення.	1	2	8	11
27. Збалансовані бінарні дерева.	1		9	10
28. Сильноразгалуджені дерева	1		8	9
Змістовий модуль 9. Хешування.	2	2	26	30
29. Принципи побудови хеш-таблиць, та їх призначення.	1		13	14
30. 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 вказників 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 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 студентами 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 ЗВО здійснюється відповідно до «Положення про поточне та підсумкове оцінювання знань здобувачів вищої освіти Національного університету «Чернігівська політехніка», погодженого вченою радою НУ «Чернігівська політехніка» (протокол № 6 від 31.08.2020 р.) та затвердженого наказом ректора НУ «Чернігівська політехніка» від 31.08.2020 р. №26.

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

Виконання та особистий захист усіх лабораторних робіт, зазначених у робочій навчальній програмі з дисципліни, є обов'язковим.

Поточний контроль проводиться шляхом спілкування зі студентами під час лекцій, лабораторних занять та консультацій, опитувань студентів під час захисту лабораторних робіт. Підсумкова оцінка поточного контролю виставляється відповідно до шкали оцінювання, що наведена у наступному розділі. Ця оцінка виставляється у відповідну графу екзаменаційної відомості.

Семестровий контроль проводиться по закінченню вивчення семестрових курсів дисципліни у вигляді іспитів. Допуск до іспиту визначається результатами поточного контролю виконання лабораторних робіт.

Якщо студент виконав лабораторні роботи лише на 70%, він має допуск до іспиту, але, у такому разі, він не може отримати оцінку, вищу за 74 бали. Студент, який не виконав лабораторні роботи на 70% та вище процентів до іспиту не допускається.

Перелік питань до іспиту знаходяться в пакеті документів на дисципліну.

Засобом оцінювання результатів навчання з дисципліни у 2-му семестрі також є курсовий проект. Оцінка за проект виставляється як середнє арифметичне двох складових за 100-бальною шкалою:

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

Політика дотримання академічної доброчесності ґрунтується на «Кодексі академічної доброчесності Національного університету «Чернігівська політехніка», погодженого вченою радою НУ «Чернігівська політехніка» (протокол №6 від 31.08.2020 р.) та введеного в дію наказом ректора НУ «Чернігівська політехніка» від 31.08.2020 р. №26.

12 Розподіл балів, які отримують студенти

Поточний контроль за модулями

Модуль за тематичним планом дисципліни та форма контролю	Кількість балів
СЕМЕСТР 1	
Змістовий модуль 1. Базові засади програмування..	0... 29
1 Повнота ведення конспектів занять.	0... 1
2 Підготовленість до лабораторних робіт.	0... 7
3 Самостійність та своєчасність виконання лабораторних робіт.	0... 7
4 Захист лабораторних робіт.	0... 14
Змістовий модуль 2. Вказівники.	0... 5
1 Повнота ведення конспектів занять.	0... 1
2 Підготовленість до лабораторних робіт.	0... 1
3 Самостійність та своєчасність виконання лабораторних робіт.	0... 1
4 Захист лабораторних робіт.	0... 2
Змістовий модуль 3. Створення та обробка простих структур даних.	0... 21
1 Повнота ведення конспектів занять.	0... 1
2 Підготовленість до лабораторних робіт.	0... 5
3 Самостійність та своєчасність виконання лабораторних робіт.	0... 5
4 Захист лабораторних робіт.	0... 10
Змістовий модуль 4. Файли.	0... 5
1 Повнота ведення конспектів занять.	0... 1
2 Підготовленість до лабораторних робіт.	0... 1
3 Самостійність та своєчасність виконання лабораторних робіт.	0... 1
4 Захист лабораторних робіт.	0... 2
Семестрова оцінка поточного контролю семестру 1	0... 60
СЕМЕСТР 2	
Змістовий модуль 5. Вказівники та зв'язані списки, мультисписки.	0... 24
1 Повнота ведення конспектів занять.	0... 4
2 Підготовленість до лабораторних робіт.	0... 4
3 Самостійність та своєчасність виконання лабораторних робіт.	0... 4
4 Захист лабораторних робіт.	0... 12
Змістовий модуль 6. Спискові структури на базі масивів вказівників.	0... 6
1 Повнота ведення конспектів занять.	0... 1
2 Підготовленість до лабораторних робіт.	0... 1
3 Самостійність та своєчасність виконання лабораторних робіт.	0... 1
4 Захист лабораторних робіт.	0... 3
Змістовий модуль 7. Абстрактні типи даних, рекурсія.	0... 18
1 Повнота ведення конспектів занять.	0... 3
2 Підготовленість до лабораторних робіт.	0... 3
3 Самостійність та своєчасність виконання лабораторних робіт.	0... 3
4 Захист лабораторних робіт.	0... 9
Змістовий модуль 8. Бінарні та сильноразгалужені дерева.	0... 6

Модуль за тематичним планом дисципліни та форма контролю	Кількість балів
1 Повнота ведення конспектів занять.	0... 1
2 Підготовленість до лабораторних робіт.	0... 1
3 Самостійність та своєчасність виконання лабораторних робіт.	0... 1
4 Захист лабораторних робіт.	0... 3
Змістовий модуль 9. Хещування.	0... 6
1 Повнота ведення конспектів занять.	0... 1
2 Підготовленість до лабораторних робіт.	0... 1
3 Самостійність та своєчасність виконання лабораторних робіт.	0... 1
4 Захист лабораторних робіт.	0... 3
Семестрова оцінка поточного контролю семестру 2	0... 60

Для захисту лабораторної роботи студент повинен відповісти на всі контрольні запитання з методичних вказівок та на два запитання за вибором викладача з лекційного курсу за темою лабораторної роботи. За кожну лабораторну роботу студент отримує певну кількість балів з урахуванням максимальної кількості 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 студент отримує за виконання лабораторних робіт. Під час семестрового контролю до визначеної суми додається оцінка за відповідь на завдання іспиту, що відображається в екзаменаційній відомості. Екзаменаційна оцінка виставляється відповідно до шкали оцінювання, наведеної в таблиці 1.

Таблиця 1 - Шкала оцінювання: національна та ECTS

Сума балів за всі види навчальної діяльності	Оцінка ECTS	Оцінка за національною шкалою	
		для екзамену, 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 медіапроектора. 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. Створення та відображення деревовидних структур даних на мові С у середовищі Qt creator. Методичні вказівки до виконання курсового проекту з дисципліни «Алгоритми та структури даних» для студентів напряму підготовки 6.050102 – „Комп'ютерна інженерія”. /Укл.: Бивойно П.Г., Борисов Д.Ю. – Чернігів: ЧНТУ, 2017. – 34 с.
4. Створення деревовидних структур даних на основі динамічних масивів вказівників. Методичні вказівки до виконання курсового проекту з дисципліни «Алгоритми та структури даних» для студентів напряму підготовки 6.050102 – „Комп'ютерна інженерія”. /Укл.: Бивойно П.Г. – Чернігів: ЧНТУ, 2017. – 33 с.
5. 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 / З.Я. Шпак. – Львів: вид-во НУ «Львівська політехніка», 2011. – 436с.
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. Керниган Б., Ритчи Д. Язык программирования С / Брайан Керниган, Деннис Ритчи – М.: Вильямс, 2013. – 304с.

5. Ахо А.В., Хопкрофт Д.Э., Ульман Д.Д. Структуры данных и алгоритмы. - М.: Вильямс, 2000. – 384с.

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 справ очник по C++. 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. Національний університет «Чернігівська політехніка». Moodle. Основи програмування (Частина 1). Режим доступу: <https://eln.stu.cn.ua/course/view.php?id=4327>
2. Національний університет «Чернігівська політехніка». Moodle. Основи програмування (Частина 2). Режим доступу: <https://eln.stu.cn.ua/course/view.php?id=3490>
3. <https://proglib.io/p/cpp-video-and-online-courses/>
4. 9 безкоштовних книг з мови Сі для початківців і не лише... – Доступу до ресурсу. : <https://echo.lviv.ua/dev/7366>