

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Дека́н ФІТ Олег САВЕНКО 2022 р.
 ЗАТВЕРДЖУЮ
 ФАКУЛЬТЕТ ПРОГРАМУВАННЯ ТА КОМП'ЮТЕРНИХ І ТЕЛЕКОМУНІКАЦІЙНИХ СИСТЕМ

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Програмування

Галузь знань 12 Інформаційні технології
 (шифр) (назва)
 Спеціальність 123 Комп'ютерна інженерія очна денна форма здобуття освіти
 (шифр) (назва)
 Освітня програма Комп'ютерна інженерія та програмування
 Шифр дисципліни ОПП.01
 Статус дисципліни: обов'язкова, дисципліна професійної підготовки
 (назва)

Факультет інформаційних технологій
 Кафедра комп'ютерної інженерії та інформаційних систем

Форма здобуття освіти	Курс	Семестр	Загальне навантаження		Кількість годин						Форма семестрового контролю			
			Кредити ЄКТС	Години	Аудиторні заняття				Індивідуальна робота студента	Самостійна робота, в т.ч. ІРС	Курсовий проект	Курсова робота	Залік	Іспит
					Разом	Лекції	Лабораторні роботи	Практичні заняття						
ОД	1	1	8	240	119	34	51	34		121			+	
ОД	1	2	6	180	72		36	36		108	+		+	
Разом			14	420	191	34	87	70		229	1		1 1	

Робоча програма складена на основі стандарту вищої освіти зі спеціальності 123 Комп'ютерна інженерія, освітньо-професійної програми та навчального плану

Програма складена Олег САВЕНКО
 Підпис Ім'я, прізвище викладача

Схвалена на засіданні кафедри комп'ютерної інженерії та інформаційних систем

Протокол № 1 від 12 08 2022 р.

Зав. кафедри комп'ютерної інженерії та інформаційних систем Тетяна ГОВОРУЩЕНКО
 Підпис Ім'я, прізвище

Робоча програма розглянута та схвалена Вченою радою факультету інформаційних технологій

Голова Вченої ради Олег САВЕНКО
 Підпис Ім'я, прізвище

Хмельницький 2022

ВСТУП

Мета викладання дисципліни. Дисципліна є однією з фундаментальних та професійних дисциплін і тому займає провідне місце у підготовці бакалаврів.

Метою дисципліни є: 1) формування компетентностей, необхідних для абстрактного мислення, аналізу та синтезу при здійсненні алгоритмізації та програмуванні; 2) розвиток у студентів фахового стилю мислення з алгоритмізації; 3) надання знань з теорії алгоритмів та конструкцій мов програмування (на прикладі мови C), необхідних для подальшого вивчення спеціальних дисциплін та для практичної інженерної діяльності; 4) вироблення у студентів вміння використовувати набуті знання при розробці програм.

Предмет дисципліни. Алгоритмізація та програмування на мові C/C++.

Завдання дисципліни. Надати студентам знання про алгоритмізацію та її використання для розв'язування практичних задач, навчити програмувати мовою C/C++.

Після вивчення дисципліни студент має досягти таких результатів навчання (сукупність знань, умінь, навичок, компетентностей):

Інтегральна компетентність: Здатність розв'язувати складні спеціалізовані задачі та практичні проблеми під час професійної діяльності в комп'ютерній галузі або навчання, що передбачає застосування теорій та методів комп'ютерної інженерії і характеризується комплексністю та невизначеністю умов.

ЗК2. Здатність вчитися і оволодівати сучасними знаннями.

ЗК3. Здатність застосовувати знання у практичних ситуаціях.

ЗК4. Здатність спілкуватися державною мовою як усно, так і письмово.

ЗК7. Вміння виявляти, ставити та вирішувати проблеми.

ЗК11. Здатність до розуміння предметної галузі та професійної діяльності.

ЗК12. Здатність використовувати інформаційні та комунікаційні технології.

ЗК13. Здатність розв'язувати поставлені задачі та приймати відповідні рішення.

ФК1. Здатність застосовувати законодавчу та нормативно-правову базу, а також державні та міжнародні вимоги, практики і стандарти з метою здійснення професійної діяльності в галузі комп'ютерної інженерії.

ФК2. Здатність використовувати сучасні методи і мови програмування для розроблення алгоритмічного та програмного забезпечення.

ФК3. Здатність створювати системне та прикладне програмне забезпечення комп'ютерних систем та мереж.

ФК17. Здатність забезпечувати проєктування та розроблення якісних програмних і технічних засобів комп'ютерних систем та мереж.

ПРН3. Знати новітні технології в галузі комп'ютерної інженерії.

ПРН6. Вміти застосовувати знання для ідентифікації, формулювання і розв'язування технічних задач спеціальності, використовуючи методи, що є найбільш придатними для досягнення поставлених цілей.

ПРН7. Вміти розв'язувати задачі аналізу та синтезу засобів, характерних для спеціальності.

ПРН10. Вміти розробляти програмне забезпечення для вбудованих і розподілених застосувань, мобільних і гібридних систем, розраховувати, експлуатувати типове для спеціальності обладнання.

ПРН11. Вміти здійснювати пошук інформації в різних джерелах для розв'язання задач комп'ютерної інженерії.

ПРН15. Вміти виконувати експериментальні дослідження за професійною тематикою.

ПРН17. Спілкуватись усно та письмово з професійних питань українською мовою та однією з іноземних мов (англійською, німецькою, італійською, французькою, іспанською).

ПРН18. Використовувати інформаційні технології для ефективного спілкування на професійному та соціальному рівнях.

ПРН19. Здатність адаптуватись до нових ситуацій, обґрунтовувати, приймати та реалізовувати у межах компетенції рішення.

ПРН20. Усвідомлювати необхідність навчання впродовж усього життя з метою поглиблення набутих та здобуття нових фахових знань, удосконалення креативного мислення.

ПРН21. Якісно виконувати роботу та досягати поставленої мети з дотриманням вимог професійної етики.

ПРН23. Використовувати базові знання інформатики й сучасних інформаційних систем та технологій, навички програмування, технології безпечної роботи в комп'ютерних мережах, методи створення баз даних та інтернет-ресурсів, технології розроблення алгоритмів і комп'ютерних програм мовами високого рівня із застосуванням об'єктно-орієнтованого програмування для розв'язання задач проектування і використання програмно-технічних засобів комп'ютерних систем та мереж.

Вміти:

складати алгоритми та програмувати їх з використанням мови C/C++, застосовувати основи програмування, вибирати парадигми програмування, складати та програмувати рекурсивні алгоритми; програмувати динамічні структури даних, обробляти стандартні і нестандартні виключення;

володіти:

основами програмування, визначати для вирішення задачі технологію програмування, мову, систему програмування, інструментальне середовище, здійснювати функціональну та об'єктну декомпозицію програми відповідно до обраної технології програмування, виконувати розробку коду програми, виправляти синтаксичні та семантичні помилки та рефакторинг коду, налагоджувати та тестувати програму, програмувати динамічні структури даних, обробляти виключення;

бути здатним:

моделювати, проектувати та розробляти алгоритми, оцінювати їх складність; планувати та ефективно організувати роботу при складанні програм мовою C/C++.

ПРОГРАМУВАННЯ

Тип дисципліни	Обов'язкова
Рівень вищої освіти	Перший (бакалаврський)
Мова викладання	Українська
Семестр	1-2
Кількість встановлених кредитів ЄКТС	14
Форма здобуття освіти	Очна денна

Результати навчання. Студент, який успішно завершив вивчення дисципліни, повинен:

ПРН3. Знати новітні технології в галузі комп'ютерної інженерії.

ПРН6. Вміти застосовувати знання для ідентифікації, формулювання і розв'язування технічних задач спеціальності, використовуючи методи, що є найбільш придатними для досягнення поставлених цілей.

ПРН7. Вміти розв'язувати задачі аналізу та синтезу засобів, характерних для спеціальності.

ПРН10. Вміти розробляти програмне забезпечення для вбудованих і розподілених застосувань, мобільних і гібридних систем, розраховувати, експлуатувати типове для спеціальності обладнання.

ПРН11. Вміти здійснювати пошук інформації в різних джерелах для розв'язання задач комп'ютерної інженерії.

ПРН15. Вміти виконувати експериментальні дослідження за професійною тематикою.

ПРН17. Спілкуватись усно та письмово з професійних питань українською мовою та однією з іноземних мов (англійською, німецькою, італійською, французькою, іспанською).

ПРН18. Використовувати інформаційні технології для ефективного спілкування на професійному та соціальному рівнях.

ПРН19. Здатність адаптуватись до нових ситуацій, обґрунтовувати, приймати та реалізовувати у межах компетенції рішення.

ПРН20. Усвідомлювати необхідність навчання впродовж усього життя з метою поглиблення набутих та здобуття нових фахових знань, удосконалення креативного мислення.

ПРН21. Якісно виконувати роботу та досягати поставленої мети з дотриманням вимог професійної етики.

ПРН23. Використовувати базові знання інформатики й сучасних інформаційних систем та технологій, навички програмування, технології безпечної роботи в комп'ютерних мережах, методи створення баз даних та інтернет-ресурсів, технології розроблення алгоритмів і комп'ютерних програм мовами високого рівня із застосуванням об'єктно-орієнтованого програмування для розв'язання задач проєктування і використання програмно-технічних засобів комп'ютерних систем та мереж.

Зміст навчальної дисципліни. Основи програмування. Парадигми програмування. Алгоритми та розв'язання задач. Фундаментальні структури даних. Структурне програмування. Конструкції мов програмування. Рекурсія. Програмування динамічних структур даних. Алгоритми та структури даних. Виключення та їх обробка.

Запланована навчальна діяльність: лекцій – 34 год., практичних занять – 70 год., лабораторних занять – 87 год., самостійної роботи – 229 год.; разом – 420 год.

Методи навчання: словесні, наочні, практичні, проблемні, інтерактивні методи, використання інформаційних технологій, частково-пошукові.

Форми оцінювання результатів навчання: усне опитування, письмові контрольні роботи, захисти лабораторних робіт, тестування.

Форма семестрового контролю: 1 семестр – іспит, 2 семестр – залік, курсовий проєкт

Навчальні ресурси:

- Шпак, З.Я. Програмування мовою C: навч. посіб. / З.Я. Шпак, Нац. ун-т "Львівська політехніка". – 2-е вид, доп. – Львів: Вид-во Львівської політехніки, 2011. – 431 с. ISBN 978-617-607-104-4.
- Татарчук Д. Д., Діденко Ю. В. Програмування мовами C та C++: навч. посіб. / Д.Д. Татарчук, Ю.В. Діденко. – К. - 2012. – 112 с.
- Матвієнко М.П. Теорія алгоритмів [Текст] : навч. посібник / М. П. Матвієнко. - Київ: Ліра-К, 2014. - 340 с.: ISBN 978-966-2609-34-9.
- Seacord R. Effective C: An Introduction to Professional C Programming. - San Francisco, California, No Starch Press, 2020. - 272 p.
- Harwani B.M. C programming cookbook: over 40 recipes exploring data structures, pointers, interprocess communication, and database in C. - Birmingham, UK, Packt Publishing, 2019. – 344 p.
- Модульне середовище для навчання MOODLE. Доступ до ресурсу: <https://msn.khnu.km.ua>.
- Електронна бібліотека університету. Доступ до ресурсу: http://lib.khnu.km.ua/asp/php_f/p1page_lib.php.

Викладач: доктор технічних наук, професор Савенко О.С.

1. Структура залікових кредитів дисципліни

Назва теми	Кількість годин, відведених на:			
	лекції	лабораторні роботи	практичні роботи	самостійну роботу
<i>Перший семестр</i>				
Тема 1. Інформаційні технології. Основи програмування. Парадигми програмування.	4	20	8	49
Тема 2. Основи програмування. Алгоритми та розв'язання задач.	4	4	8	7
Тема 3. Основи програмування. Фундаментальні структури даних.	6	4	8	7
Тема 4. Структурне програмування. Конструкції мов програмування.	14	23	10	58
Тема 5. Програмування динамічних структур даних. Алгоритми та структури даних.	6			
Разом за 1-ий семестр:	34	51	34	121
<i>Другий семестр</i>				
Тема 6. Програмування динамічних структур даних. Алгоритми та структури даних.		8	12	48
Тема 7. Об'єктно-орієнтоване програмування. Парадигми об'єктно-орієнтованого програмування. Обробка виключень. Виключення та їх обробка.		28	24	60
Разом за 2-ий семестр:		36	36	108

2. ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

2.1 Зміст лекційного курсу

Номер лекції	Перелік змістових модулів, тем лекцій, їх анотації	Кількість годин
<i>Перший семестр</i>		
	Тема 1. Інформаційні технології. Основи програмування. Парадигми програмування.	
1	Лекція 1. Вступ. Інформаційні технології Вступ. Інформація та її подання. Двійкова арифметика. Літ. [15, С.14-40]	2
2	Лекція 2. Вступ. Комп'ютерні системи Загальна структура комп'ютера. Комп'ютерні системи та їх складові. Прикладне програмне забезпечення. Системне програмне забезпечення. Парадигми програмування. Літ. [15, С.14-40]	2
	Тема 2. Основи програмування. Алгоритми та розв'язання задач.	
3	Лекція 3. Алгоритми та їх властивості. Основні поняття Алгоритми і алгоритмізація. Поняття алгоритму. Алгоритми та їх властивості. Поняття про алгоритми та їх властивості. Властивості алгоритму. Метод покрокової деталізації. Форми подання алгоритмів. Літ. [3, С.12-27]	2
4	Лекція 4. Алгоритми та їх властивості. Типи алгоритмів Запис алгоритму у вигляді блок-схем. Приклади. Базові алгоритмічні структури. Слідування. Розгалуження. Повторення. Приклади. Типи основних структур алгоритмів. Поняття програми. Літ. [3, С.12-27]	2
	Тема 3. Основи програмування. Фундаментальні структури даних.	
5	Лекція 5. Введення в С Вступ до мови програмування С. Алфавіт та словник мови. Ідентифікатори. Базові типи даних. Модифікатори типів. Літ. [1, С.7-18; 2, С. 6-12; 5-14]	2
6	Лекція 6. Структура програми мови С Константи. Префіксна та суфіксна форми. Вирази та операції. Десятькове, вісімкове та шістнадцяткове представлення. Ідентифікатори. Ключові слова. Коментарі. Стандарти мови С. Літ. [1, С.7-18; 2, С. 13-20; 5-14]	2
7	Лекція 7. Структура програми мови С. Стандартні функції Структура програми мови Паскаль. Прототипи функцій. Директиви для включення вмісту файлів. Основні засоби введення-виведення. Специфікації виведення функцій. Введення	2

	даних з використанням стандартних функцій. Літ. [1, С.19-30; 2, С. 27-48; 5-14]	
	Тема 4. Структурне програмування. Конструкції мов програмування.	
8	Лекція 8. Загальні відомості про оператори Загальні відомості про оператори. Складений оператор. Порожній оператор. Вирази. Знак операції. Типи арифметичних операцій. Пріоритет операцій. Літ. [1, С.31-111; 2, С. 17-26; 5-14]	2
9	Лекція 9. Загальні відомості про оператори Оператор розгалуження, його форми. Логічні операції. Складені оператори присвоювання. Операція вибору за умовою. Літ. [1, С.85-111; 2, С. 17-27; 5-14]	2
10	Лекція 10. Загальні відомості про оператори Використання коми. Оператори циклу. Оператор з параметром та його особливості. Оператори циклу-поки та циклу-до, їх відмінності. Літ. [1, С.85-111; 2, С. 17-27; 4-14]	2
11	Лекція 11. Функції в мові С Оголошення і визначення функцій. Аргументи, параметри, приклади. Передача параметрів за значенням та за посиланням. Повернення з функції та повернення значень. Арифметичні функції. Рекурсія. Рекурсивні виклики. Пряма та непряма рекурсії. Основні принципи структурного програмування. Різні типи функцій. Літ. [1, С.183-234; 2, С. 27-47; 4-14]	2
12	Лекція 12. Масиви Опис масивів. Масиви елементів. Оголошення масивів. Доступ до компонентів масиву, одно- і n-мірні масиви. Приклади. Методи сортування та пошуку. Обчислювальна складність алгоритмів. Літ. [1, С.125-140; 2, С. 29; 3, С.44-59, 105-117; 4-14]	2
13	Лекція 13. Рядки Рядки. Поняття та оголошення. Індeksi елементів рядка. Об'єднання рядків. Порівняння рядків. Введення – виведення рядків. Інші функції для обробки рядків. Приклади. Літ. [1, С.141-163; 2, С. 52-53; 4-14]	2
14	Лекція 14. Файли Файли. Визначення. Типи файлів. Загальні правила для всіх типів файлів. Текстові файли. Алгоритм створення текстового файлу, читання, запис, дозапис в текстовий файл. Особливості застосування стандартних функцій і процедур при роботі з файлами. Літ. [1, С.311-362; 2, С. 54-60; 4-14]	2

	Тема 5. Програмування динамічних структур даних. Алгоритми та структури даних.	
15	Лекція 15. Вказівники. Поняття вказівників. Створення вказівників. Оголошення вказівників. Вказівники і типи змінних. Вказівники і масиви. Передача масивів в функції. Динамічні структури даних. Літ. [1, С.112-124; 2, С. 34-35; 4-14]	2
16	Лекція 16. Структури, об'єднання і нестандартні типи даних. Найпростіші структури. Складні структури. Масиви структур. Літ. [1, С.112-124; 2, С. 30-31; 4-14]	2
17	Лекція 17. Додаткові відомості про вказівники Вказівники на вказівники. Відмінність динамічних даних від статичних. Стек, черга, двійкове дерево. Літ. [1, С.246-295; 2, С. 35; 4-14]	2
	Разом за 1-ий семестр:	34

2.2 Зміст лабораторних занять

№ з/п	Тема лабораторного заняття	Кількість годин
<i>Перший семестр</i>		
1	Вступ до робототехніки. Lego Mindstorms EV3. Прямолінійний рух, повороти, розворот на місці та зупинка. Програмування лінійних алгоритмів.	4
2	Знайомство з набором датчиків в Lego Mindstorms EV3. Керуючі структури.	4
3	Знайомство з масивами в LEGO Mindstorms EV3. Зв'язок роботів через Bluetooth.	4
4	Основи проведення змагань роботів LEGO Mindstorms EV3.	4
5	Лінійні алгоритми. Реалізація мовою С.	4
6	Розгалужені алгоритми. Реалізація мовою С.	4
7	Програмування циклічних алгоритмів. Реалізація мовою С.	4
8	Масиви. Робота з одномірними масивами. Реалізація мовою С.	4
9	Двомірні масиви. Реалізація мовою С.	4
10	Функції. Реалізація мовою С.	4
11	Рядки. Реалізація мовою С.	4
12	Потоки та файли. Робота з текстовими файлами. Їх реалізація мовою С.	4
13	Підсумкове заняття.	3
Разом за 1-ий семестр:		51
<i>Другий семестр</i>		
1	Бінарні файли. Реалізація мовою С.	4
2	Динамічні структури даних. Однозв'язні та двозв'язні списки.	4
3	Створення проекту додатка Windows Forms. Програмування елементів керування.	4
4	Робота з діалоговими компонентами та файлами.	4
5	Розробка програмних додатків із використанням графіки.	4
6	Динамічні структури даних. Черга, циклічна черга.	4
7	Динамічні структури даних. Стек.	4
8	Динамічні структури даних. Двійкові дерева.	4
9	Підсумкове заняття.	4
Разом за 2-ий семестр:		36

2.3 Зміст практичних занять

№ з/п	Тема практичного заняття	Кількість годин
<i>Перший семестр</i>		
1	<i>Практичні роботи №1-2.</i> Лінійні алгоритми.	4
2	<i>Практичні роботи №3-4.</i> Розгалужені алгоритми.	4
3	<i>Практичні роботи №5-6.</i> Програмування циклічних алгоритмів.	4
4	<i>Практичні роботи №7-8.</i> Масиви. Робота з одновимірними масивами.	4
5	<i>Практичні роботи №9-10.</i> Двовимірні масиви.	4
6	<i>Практичні роботи №11-12.</i> Функції.	4
7	<i>Практичні роботи №13-14.</i> Рядки.	4
8	<i>Практичні роботи №15-16.</i> Потоки та файли. Робота з текстовими файлами. Структури.	4
9	<i>Практична робота №17.</i> Підсумкове заняття.	2
	Разом за 1-ий семестр:	34
<i>Другий семестр</i>		
1	<i>Практичні роботи №1-2.</i> Бінарні файли.	4
2	<i>Практичні роботи №3-4.</i> Динамічні структури даних. Однозв'язні списки.	4
3	<i>Практичні роботи №5-6.</i> Динамічні структури даних. Двозв'язні списки.	4
4	<i>Практичні роботи №7-8.</i> Динамічні структури даних. Черга. Циклічна черга.	4
5	<i>Практичні роботи №9-10.</i> Динамічні структури даних. Стек.	4
6	<i>Практичні роботи №11-12.</i> Створення головного меню додатку, контекстного меню, спливаючої підказки.	4
7	<i>Практичні роботи №13-14.</i> Створення багато віконного додатку.	4
8	<i>Практичні роботи №15-16.</i> Обробка виключень.	4
9	<i>Практичні роботи №17-18.</i> Розробка бібліотек динамічної компоновки (DLL).	4
	Разом за 2-ий семестр:	36

2.4 Зміст самостійної (індивідуальної) роботи

Обсяг самостійної роботи з дисципліни “Програмування” в I семестрі становить 121 годин. Він включає опрацювання лекційного матеріалу, підготовку до виконання лабораторних робіт і їх захисту, підготовку до виконання практичних робіт, підготовку до поточного контролю.

Обсяг самостійної роботи з дисципліни “Програмування” в II семестрі становить 108 годин. Він включає підготовку до виконання практичних та лабораторних робіт і їх захисту, підготовку до поточного контролю, виконання і захист курсового проекту на 16-18-ому тижнях семестру.

Керівництво самостійною роботою та виконанням курсового проекту здійснює викладач згідно з розкладом консультацій в позаурочний час.

Номер тижня	Вид самостійної роботи	К-ть годин
<i>Перший семестр</i>		
1	Опрацювання лекційного матеріалу. Підготовка до лабораторної роботи №1 та практичної роботи №1. Самостійна робота над розробкою програми до лабораторної роботи №1.	7
2	Опрацювання лекційного матеріалу. Підготовка до лабораторної роботи №2 та практичної роботи №2. Самостійна робота над розробкою програми до лабораторної роботи №2.	7
3	Опрацювання лекційного матеріалу. Підготовка до лабораторної роботи №3 та практичної роботи №3. Самостійна робота над розробкою програми до лабораторної роботи №3.	7
4	Опрацювання лекційного матеріалу. Підготовка до лабораторної роботи №3 та практичної роботи №4. Самостійна робота над розробкою програми до лабораторної роботи №3.	7
5	Опрацювання лекційного матеріалу. Підготовка до лабораторної роботи №4 та практичної роботи №5. Самостійна робота над розробкою програми до лабораторної роботи №4.	7
6	Опрацювання лекційного матеріалу. Підготовка до лабораторної роботи №5 та практичної роботи №6. Самостійна робота над розробкою програми до лабораторної роботи №5.	7
7	Опрацювання лекційного матеріалу. Підготовка до лабораторної роботи №6 та практичної роботи №7. Самостійна робота над розробкою програми до лабораторної роботи №6.	7
8	Опрацювання лекційного матеріалу. Підготовка до лабораторної роботи №6 та практичної роботи №8. Самостійна робота над розробкою програми до лабораторної роботи №6.	7
9	Опрацювання лекційного матеріалу. Підготовка до лабораторної роботи №7 та практичної роботи №9. Самостійна робота над розробкою програми до лабораторної роботи №7.	7
10	Опрацювання лекційного матеріалу. Підготовка до лабораторної роботи №8 та практичної роботи №10. Самостійна робота над розробкою програми до лабораторної роботи №8.	7
11	Опрацювання лекційного матеріалу. Підготовка до лабораторної роботи №9 та практичної роботи №11. Самостійна робота над розробкою програми до лабораторної роботи №9.	7
12	Опрацювання лекційного матеріалу. Підготовка до лабораторної роботи №10 та практичної роботи №12. Самостійна робота над розробкою програми до лабораторної роботи №10.	7

13	Опрацювання теоретичного матеріалу. Підготовка до лабораторної роботи №7 та практичної роботи №13. Самостійна робота над розробкою програми до лабораторної роботи №7.	6
14	Опрацювання теоретичного матеріалу. Підготовка до лабораторної роботи №7 та практичної роботи №14. Самостійна робота над розробкою програми до лабораторної роботи №7.	6
15	Опрацювання теоретичного матеріалу. Підготовка до лабораторної роботи №8 та практичної роботи №15. Самостійна робота над розробкою програми до лабораторної роботи №8.	6
16	Опрацювання теоретичного матеріалу. Підготовка до лабораторної роботи №8 та практичної роботи №16. Самостійна робота над розробкою програми до лабораторної роботи №8.	6
17	Опрацювання теоретичного матеріалу. Підготовка до практичної роботи №17.	6
18	Підготовка до залікового заняття №18.	6
Разом за 2-ий семестр:		108

Тематика курсового проекту (II семестр)

1. Розробка діалогової або тестової програми.
2. Розробка програми вводу і обробки файлу з друком вихідної форми.
3. Розробка навчальних тестових програм з інтерфейсами сучасних систем.

3. МЕТОДИ НАВЧАННЯ

Процес навчання з дисципліни ґрунтується на використанні традиційних та сучасних методів. Зокрема, використовуються методи: словесні, наочні, практичні, проблемні, інтерактивні методи, використання інформаційних технологій, частково-пошукові.

4. ОЦІНЮВАННЯ РЕЗУЛЬТАТІВ НАВЧАННЯ СТУДЕНТІВ

Поточний контроль здійснюється під час лекційних, практичних та лабораторних занять, а також у дні проведення контрольних заходів, встановлених робочим планом дисципліни. Семестрові контролі проводяться у формі іспиту, заліку та захисту курсового проекту. При цьому при виведенні остаточної оцінки враховуються результати поточного контролю.

При викладанні дисципліни використовуються такі види навчальних занять, як лекції, лабораторні роботи, практичні роботи, курсове проектування, індивідуальне консультування і керівництво самостійною роботою студента.

Кожний, обов'язковий для оцінювання, вид роботи з дисципліни оцінюється за *чотирибальною* шкалою. Семестрова підсумкова оцінка визначається як середньозважена з усіх видів навчальної роботи, виконаних і зданих *позитивно* з врахуванням коефіцієнта вагомості. Вагові коефіцієнти змінюються залежно від структури дисципліни і важливості окремих її видів робіт. Студент, який набрав позитивний середньозважений бал за поточну роботу і не здав підсумковий контрольний захід (іспит), вважається невстигаючим.

При оцінюванні знань студентів використовуються різні засоби контролю, зокрема: усне опитування перед допуском до виконання лабораторної роботи – здійснюється на її початку; засвоєння теоретичного матеріалу з тем перевіряється тестовим контролем; якість виконання, набуття теоретичних знань і практичних навичок перевіряється шляхом захисту кожної лабораторної роботи згідно з робочою програмою дисципліни і робочим навчальним планом.

Оцінка, яка виставляється за *лабораторне заняття*, складається з таких елементів: усне опитування студентів перед допуском до виконання лабораторної роботи; знання теоретичного матеріалу з теми; якість оформлення протоколу і графічної частини; вміння студента обґрунтувати прийняті конструктивні рішення; своєчасний захист лабораторної роботи. Для виконання програми дисципліни студент повинен отримати 12 (осінній семестр) і 8 (весняний семестр) оцінок за лабораторні роботи в кожному з семестрів.

Термін захисту лабораторної роботи вважається своєчасним, якщо студент захистив її на наступному після виконання роботи занятті.

Пропущене лабораторне заняття студент повинен відпрацювати в лабораторіях кафедри у встановлений викладачем термін з реєстрацією у відповідному журналі кафедри, але не пізніше, ніж за два тижні до кінця теоретичних занять у семестрі.

При *оцінюванні знань* студентів викладач керується такими критеріями.

Оцінку „відмінно”, за шкалою ECTS – A (див. шкалу оцінок), отримує студент за глибоке і повне опанування змісту навчального матеріалу, в якому він легко орієнтується, понятійного апарату, за уміння зв’язувати теорію з практикою, вирішувати практичні завдання, висловлювати і обґрунтовувати свої судження. Відмінна оцінка передбачає грамотний, логічний виклад відповіді (як в усній, так і в письмовій формі), якісне зовнішнє оформлення. Студент повинен набути практичних навичок із складання різних алгоритмів та розробки програм за цими алгоритмами. Оцінка "відмінно" виставляється студенту, який глибоко засвоїв оператори, функції та процедури мови C та вміє їх раціонально застосувати, знає методики та вміє ними користуватися при складанні алгоритмів та програм. Студент не повинен вагатися при видозміні запитання, повинен робити детальні та узагальнюючі висновки.

Оцінку „добре”, за шкалою ECTS – B, отримує студент за повне засвоєння навчального матеріалу, володіння понятійним апаратом, орієнтування в вивченому матеріалі, свідоме використання знань для вирішення практичних завдань, грамотний виклад відповіді, але у змісті і формі відповіді мали місце окремі неточності (похибки), нечіткі формулювання закономірностей тощо. Відповідь студента повинна будуватись на основі самостійного мислення.

Оцінку „добре”, за шкалою ECTS – C, отримує студент за правильну відповідь з однією суттєвою помилкою.

Оцінки "задовільно", за шкалою ECTS – D, заслуговує студент, який виявив знання основного навчально-програмного матеріалу в обсязі, необхідному для подальшого навчання та практичної діяльності за професією, що справляється з виконанням практичних завдань, передбачених програмою. Як правило, відповідь студента будується на рівні репродуктивного мислення, студент слабо знає структуру курсу, допускає помилки у відповіді, засвоїв і набув практичних навичок у складанні програм, але допустив неточності. Вагається при відповіді на видозмінене запитання, разом з тим студент володіє знаннями, що дозволяють йому під керівництвом викладача усунути неточності у відповіді.

Оцінки "задовільно", за шкалою ECTS – E, заслуговує студент за неповне опанування програмного матеріалу, але ним отримані знання і набуті практичні навички із розробки програм мовою C.

Оцінка „незадовільно”, за шкалою ECTS – FX, виставляється, коли студент має розрізнені, безсистемні знання, не вміє виділяти головне і другорядне, допускається помилок у визначенні понять, перекручує їх зміст, хаотично і невпевнено викладає матеріал, не може використовувати знання при вирішенні практичних завдань. Як правило, оцінка "незадовільно" виставляється студенту, який не може продовжити навчання без додаткових знань з курсу.

Оцінка „незадовільно”, за шкалою ECTS – F, виставляється студенту за повне незнання і нерозуміння навчального матеріалу або відмову від відповіді і передбачає повторне навчання студента з дисципліни.

Кожний вид роботи оцінюється за чотирибальною шкалою. Семестрова підсумкова оцінка визначається як середньозважена з усіх видів робіт.

Структурування дисципліни за видами робіт і оцінювання результатів навчання студентів у семестрі за ваговими коефіцієнтами

Аудиторна робота												Самостійна, індивідуальна робота						Форма семестрового контролю					
I семестр																							
Лабораторні роботи №:				Практичні роботи №					Тестовий контроль:			Контрольні роботи			Іспит								
1,2	3,4	5,6	7,8	9	10	11	12	1	2	3	4	5	6	7	8	9	T 1-17	T 1-8	T 9 - 12	T 13 - 17			
ВК: 0,2				ВК: 0					0,1			0,1			0,1			0,1			0,4		
II семестр																							
Лабораторні роботи №:				Практичні роботи №											Залік								
1	2	3	4	5	6	7	8	1,2	3,4	5,6	7,8	9,10	11,12	13,14	15,16	17,18							
ВК: 1				ВК: 0											0								
II семестр (курсний проект)																							
Контрольні точки №												Програма			Креслення			Захист КП					
1 розділ			2 розділ			3 розділ																	
ВК: 0,1			ВК: 0,2			ВК: 0,2						ВК: 0,2			ВК: 0,1			0,2					

Примітка: Т – тема дисципліни; ВК – ваговий коефіцієнт;

Оцінювання тестових завдань. Тематичний тест для кожного студента складається з двадцяти п'яти тестових завдань, кожне з яких оцінюється одним балом. Максимальна сума балів, яку може набрати студент, складає 25.

Оцінювання здійснюється за чотирибальною шкалою.

Відповідність набраних балів за тестове завдання оцінці, що виставляється студенту, представлена у нижченаведеній таблиці.

Сума балів за тестове завдання	1–11	12–14	15–22	23-25
Оцінка	2	3	4	5

На тестування відводиться 30 хвилин. Тестування проводиться з використанням модульного середовища для навчання MOODLE. Правильні відповіді студент реєструє в он-лайн режимі в модульному середовищі MOODLE. Через 30 хвилин студенти завершують тестування та надсилають свої відповіді на сервер. Викладач оголошує результати тестування згідно журналу оцінок модульного середовища MOODLE.

Якщо студент отримав негативну оцінку, то він має перездати її в установленому порядку, але обов'язково до терміну наступного контролю.

Підсумкова семестрова оцінка за національною шкалою і шкалою ЄКТС встановлюється в автоматизованому режимі після внесення усіх оцінок до електронного журналу. Співвідношення вітчизняної шкали оцінювання і шкали оцінювання ЄКТС наведені у наступній таблиці.

Для переходу від вітчизняної оцінки до оцінки за шкалою ECTS необхідно знайти середньоарифметичну оцінку за вітчизняною шкалою, помножити її на відповідний ваговий коефіцієнт і, додавши всі складові, отримаємо суму балів, які визначають конкретну оцінку ECTS.

Співвідношення вітчизняної шкали оцінювання і шкали оцінювання ЄКТС

Оцінка ECTS	Бали	Вітчизняна оцінка	
A	4,75-5,00	5	ВІДМІННО – глибоке і повне опанування навчального матеріалу і виявлення відповідних умінь та навиків
B	4,25-4,74	4	ДОБРЕ – повне знання навчального матеріалу з кількома незначними помилками
C	3,75-4,24	4	ДОБРЕ – в загальному правильна відповідь з однією суттєвою помилкою
D	3,25-3,74	3	ЗАДОВІЛЬНО – неповне опанування програмного матеріалу, але достатнє для практичної діяльності за професією
E	3,00-3,24	3	ЗАДОВІЛЬНО – неповне опанування програмного

			матеріалу, що задовольняє мінімальні критерії оцінювання
FX	2,00 -2,99	2	НЕЗАДОВІЛЬНО – безсистемність одержаних знань і неможливість продовжити навчання без додаткових знань з дисципліни
F	0,00-1, 99	2	НЕЗАДОВІЛЬНО – необхідна серйозна подальша робота і повторне вивчення дисципліни

Залік виставляється при отриманні студентом з дисципліни від 3,00 до 5,00 балів. При цьому за вітчизняною шкалою ставиться «зараховано», а за шкалою ECTS – оцінка, що відповідає набраній студентом кількості балів.

5. ПИТАННЯ ДЛЯ САМОКОНТРОЛЮ СТУДЕНТІВ

1. Інформаційні технології та системи
2. Історія розвитку комп'ютерної техніки
3. Історія розвитку операційних систем, їх основні поширені реалізації
4. Інформація та її подання
5. Двійкова арифметика
6. Загальна структура комп'ютера
7. Комп'ютерні системи та їх складові
8. Поняття про алгоритми та їх властивості
9. Метод покрокової деталізації
10. Запис алгоритму у вигляді блок-схем
11. Форми подання алгоритмів
12. Типи основних структур алгоритмів
13. Алфавіт та словник мови C
14. Базові типи даних
15. Константи
16. Ідентифікатори
17. Ключові слова
18. Коментарі
19. Стандарти мови C
20. Структура програми мови C
21. Директива `#include`
22. Основні засоби введення-виведення
23. Загальні відомості про оператори
24. Вирази
25. Оператор розгалуження
26. Логічні операції
27. Складені оператори присвоювання
28. Операція вибору за умовою
29. Використання коми
30. Оператори циклу
31. Цикл з післяумовою, приклад програми
32. Цикл FOR
33. Цикл з передумовою, приклад програми
34. Оголошення і визначення функцій
35. Рекурсивний виклик
36. Масиви елементів
37. Багатомірні масиви
38. Метод попарної перестановки елементів
39. Метод найменших елементів
40. Метод бінарного пошуку
41. Поняття вказівників
42. Створення вказівників
43. Оголошення вказівників
44. Вказівники і типи змінних
45. Вказівники і масиви
46. Передача масивів в функції

47. Поняття про рядки
48. Способи виділення пам'яті
49. Функції введення і виведення символів та рядків
50. Операції інкременту та декременту
51. Оператор continue
52. Оператор break
53. Повернення значень функції за допомогою оператора return
54. Прототипи
55. Додаткові операції присвоювання
56. Файли об'єктного коду, виконувані файли і бібліотеки
57. Оператор розгалуження IF
58. Складені літерали.
59. Взаємодія з файлами.
60. Стандартні файли.
61. Функція fopen().
62. Програма зтискування файлів.
63. Введення – виведення файлів.
64. Довільний доступ до файлів
65. Найпростіші структури даних
66. Складні структури.
67. Масиви структур.
68. Ініціалізація структур.

6. МЕТОДИЧНЕ ЗАБЕЗПЕЧЕННЯ

Навчальний процес з дисципліни «Програмування» повністю і в достатній кількості забезпечений необхідною навчально-методичною літературою. Зокрема, викладачами кафедри підготовлені і видані:

1. **Програмування.** Методичні вказівки і завдання для виконання лабораторних робіт для студентів напряму «Комп'ютерна інженерія»/ О.С. Савенко, Ю.П. Кльоц, С.В. Мостовий, С.М. Лисенко. – Хмельницький: ХНУ, 2015. – 134 с.
2. **Програмування.** Методичні вказівки до виконання лабораторних робіт для студентів спеціальності «Комп'ютерна інженерія» / О.С. Савенко, А.О. Нічепорук, К.Ю. Бобровнікова – Хмельницький: ХНУ, 2016. – 136 с.
3. **Програмування.** Методичні вказівки до курсового проектування з навчальної дисципліни «Програмування» для студентів спеціальностей «Комп'ютерна інженерія» та «Інформаційні системи та технології» / Савенко О.С., Бобровнікова К.Ю., Нічепорук А.О., Медзятий Д.М., 2019. – 62 с.
4. С.М. Лисенко **Програмування робототехнічних систем на основі Lego Mindstorms** / Лисенко С.М., Нічепорук А.О., Бобровнікова К.Ю. // Хмельницький, ХНУ, 2020. – 242 с. ISBN 978-966-330-367-3.

7. РЕКОМЕНДОВАНА ЛІТЕРАТУРА

Основна література

1. Шпак, З Я. Програмування мовою С: навч. посіб. / З.Я. Шпак, Нац. ун-т "Львівська політехніка". – 2-е вид, доп. – Львів: Вид-во Львівської політехніки, 2011. – 431 с. ISBN 978-617-607-104-4.
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. Матвієнко М.П. Теорія алгоритмів [Текст] : навч. посібник / М. П. Матвієнко. - Київ: Ліра-К, 2014. - 340 с.: ISBN 978-966-2609-34-9.
4. Seacord R. Effective C: An Introduction to Professional C Programming. - San Francisco, California, No Starch Press, 2020. - 272 p.
5. Harwani B.M. C programming cookbook: over 40 recipes exploring data structures, pointers, interprocess communication, and database in C. - Birmingham, UK, Packt Publishing, 2019. – 344 p.
6. Joyce P. Numerical C: applied computational programming with case studies. - Berkeley, CA, Apress, 2019. – 312 p.
7. Gustedt J. Modern C. - Shelter Island, Manning, 2020. – 408 p.

8. Stephen Prata. C Primer Plus (Developer's Library) 6th Edition. - Addison-Wesley Professional, 2013. - 1072 p.
9. McGrath M. C Programming in easy steps: Updated for the GNU Compiler version 6.3.0 and Windows 10 5th Edition. - In Easy Steps Limited, 2018. – 192 p.
10. King K.N. C programming: a modern approach. - W. W. Norton & Company, 2020. – 832 p.
11. Harwani B.M. Practical C Programming: Solutions for modern C developers to create efficient and well-structured programs. – Birmingham, Packt Publishing, 2020. – 616 p.
12. Hubert H.W. Intermediate C programming for the PIC microcontroller: simplifying embedded programming. - Berkeley, CA, Apress L. P., 2020. – 318 p.
13. Winkle L.Van. Hands-On Network Programming with C: Learn socket programming in C and write secure and optimized network code. – Birmingham, Packt Publishing, 2019. – 478 p.
14. Paul Deitel, Harvey Deitel. C for Programmers with an Introduction to C11. – Deitel, 2013. - 480 p.
15. Гірінова, Л.В. Інформаційні системи та технології. Частина 1. Технічне та програмне забезпечення інформаційних технологій та систем:: навч. посібник / Л.В. Гірінова, І.Г. Сибірякова. – Харків: Monograf, 2016. – 121 с.

Додаткова література

1. Kane W, C programming for business. Book one, Introduction to problem solving in C language. - Choice Publishing, 2017. – 284 p.
2. Стандарт C11 <http://www.open-std.org/jtc1/sc22/wg14/www/docs/n1570.pdf>

Інформаційні ресурси

Електронний університет:

1. Модульне середовище для навчання (розміщені усі необхідні матеріали з дисципліни, в тому числі тестові завдання для поточного контролю знань та типові варіанти письмових контрольних робіт).
2. Електронна бібліотека університету.