

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ імені ІВАНА ФРАНКА

Кафедра (циклова комісія) _____ системного проектування _____

“ЗАТВЕРДЖУЮ”

Декан факультету _____

доц. Юрій ФУРГАЛА _____

“

”

2022 року

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

ОСНОВИ ПРОГРАМУВАННЯ

(шифр і назва навчальної дисципліни)

спеціальність _____ 121 – Інженерія програмного забезпечення _____

(шифр і назва спеціальності)

спеціалізація _____

(назва спеціалізації)

факультет _____ електроніки та комп'ютерних технологій _____

(назва інституту, факультету, відділення)

Робоча програма “Основи програмування” для студентів
(назва навчальної дисципліни)
 галузі знань “12 – Інформаційні технології”
 за спеціальністю “121 Інженерія програмного забезпечення”

Розробники: Іван КАТЕРИНЧУК (кандидат фізико-математичних наук, доцент,
доцент кафедри оптоелектроніки та інформаційних технологій);
Олег КРУПИЧ (доктор фізико-математичних наук, ст.н.сп.,
професор кафедри оптоелектроніки та інформаційних технологій)
(вказати авторів, їхні посади, наукові ступені та вчені звання)

Робочу програму схвалено на засіданні кафедри (циклової комісії) _____
оптоелектроніки та інформаційних технологій

Протокол від “ 29 ” 08 2022 року № 9

Завідувач кафедри оптоелектроніки та ІТ


(підпис)

Олег КУШНІР
(прізвище та ініціали)

Ухвалено Вченою радою факультету електроніки та комп'ютерних технологій

Протокол від “ 31 ” 08 2022 року № 28/22

© Катеринчук І., 2022 рік

© Крупич О., 2022 рік

© ЛНУ ім. І. Франка, 2022 рік

1. Опис навчальної дисципліни

Найменування показників	Галузь знань, освітньо-кваліфікаційний рівень	Характеристика навчальної дисципліни	
		денна форма навчання	заочна форма навчання
Кількість кредитів 5	Галузь знань <u>12 Інформаційні технології</u> (шифр і назва)	Нормативна	
Модулів – <i>1</i>	Спеціальність: <u>121 Інженерія програмного забезпечення</u>	Рік підготовки	
Змістових модулів – 2		1-й	
Індивідуальне науково-дослідне завдання <u>немає</u> (назва)		Семестр	
Загальна кількість годин – 150		1-й	
Тижневих годин для денної форми навчання: аудиторних – 4 самостійної роботи студента – 3,5	Освітній ступінь <u>бакалавр</u>	Лекції	
		32 год.	
		Практичні, семінарські	
		<i>немає</i>	
		Лабораторні	
		32 год.	
		Самостійна робота	
		86 год.	
		Індивідуальні завдання:	
		<i>немає</i>	
Вид контролю:			
<i>екзамен</i>			

Примітка.

Співвідношення кількості годин аудиторних занять до самостійної і індивідуальної роботи становить

для денної форми навчання – 0,74

для заочної форми навчання – немає

2. Мета та завдання навчальної дисципліни

Мета: формувати в студентів здатність до абстрактного мислення, аналізу та синтезу, а також до алгоритмічного та логічного мислення; надати студентам необхідні теоретичні та практичні знання з алгоритмізації та програмування; сформувати практичні навички, які б дали змогу ефективно застосовувати знання з основ програмування.

Цілі: ознайомити студентів з поняттями про: основи програмування, абстракцію та алгоритми, компіляцію та інтерпретацію, середовище розробки, ключові слова, ідентифікатори та модифікатори, типи даних, прості оператори,

оператори галуження та цикли, функції, масиви одномірні та багатомірні, стрічки та роботу з ними, структуру пам'яті Сі програми, вказівники, структури, систему вводу / виводу, робота з файловою системою, директиви.

У результаті вивчення навчальної дисципліни студент повинен

знати: актуальність вивчення алгоритмів та мов програмування, методи представлення алгоритмів, способи програмної реалізації алгоритмів та засоби відлагодження програм;

вміти: класифікувати та розв'язувати складні спеціалізовані завдання інженерії програмного забезпечення, пов'язані зі створенням та програмною реалізацією алгоритмів, вміти застосовувати знання основ програмування в процесі практичного написання програмного коду, використовувати сучасні середовища розробки під час створення та відлагодження програм різного призначення.

Після вивчення даного курсу здобувачі набудуть таких Загальних та Фахових компетентностей та Програмних результатів навчання:

K01. Здатність до абстрактного мислення, аналізу та синтезу.

K02. Здатність застосовувати знання у практичних ситуаціях.

K03. Здатність спілкуватися державною мовою як усно, так і письмово.

K04. Здатність спілкуватися іноземною мовою як усно, так і письмово.

K11. Здатність реалізувати свої права і обов'язки як члена суспільства, усвідомлювати цінності громадянського (вільного демократичного) суспільства та необхідність його сталого розвитку, верховенства права, прав і свобод людини і громадянина в Україні.

K18. Здатність розробляти архітектури, модулі та компоненти програмних систем.

K23. Здатність застосовувати фундаментальні і міждисциплінарні знання для успішного розв'язання завдань інженерії програмного забезпечення.

K25. Здатність накопичувати, обробляти та систематизувати професійні знання щодо створення і супроводження програмного забезпечення та визнання важливості навчання протягом всього життя.

K26. Здатність реалізовувати фази та ітерації життєвого циклу програмних систем та інформаційних технологій на основі відповідних моделей і підходів розробки програмного забезпечення.

K28. Здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення.

K29. Здатність до алгоритмічного та логічного мислення.

K30. Здатність використовувати для розробки програмного забезпечення перспективні технології, зокрема, системи штучного інтелекту, Fog/Edge-обчислення тощо.

K31. Володіння методами, серверними технологіями та інструментальними засобами проектування веб-застосувань.

K32. Здатність здійснювати розробку програмного забезпечення використовуючи різні парадигми програмування (в тому числі паралельне, об'єктно-орієнтоване, функціональне програмування тощо)

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.

ПР13. Знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань.

ПР17. Вміти застосовувати методи компонентної розробки програмного забезпечення.

3. Програма навчальної дисципліни

Змістовий модуль 1. *Основи програмування та алгоритмізації.*

Тема 1. *Вступ до курсу основи програмування (2 год.).*

1. Вступ до курсу.
2. Актуальність вивчення курсу «Основи програмування».
3. Історичні аспекти розвитку програмування та комп'ютерної техніки.
4. Поняття про комп'ютер, інформацію, дані.
5. Зміст етапів інформаційного процесу.
6. Комп'ютерні науки.
7. Історія виникнення мови Сі.
8. Місце Сі серед інших мов програмування.

Тема 2. *Абстракція та алгоритми (2 год.).*

9. Поняття абстракції.
10. Двійкова/шіснадцяткова система числення.
11. ASCII таблиця.
12. Принципи побудови зображення, статичного та динамічного.
13. Поняття алгоритму та його ефективності.
14. Алгоритм половинного ділення.
15. Способи представлення алгоритмів.
16. Абстракція, псевдокод.

Тема 3. *Компіляція та інтерпретація. Середовище розробки (2 год.).*

17. Компіляція та інтерпретація.
18. Принципи побудови і структура проекту.
19. Середовище Scath.
20. Середовище Visual Studio.
21. Середовище Dev-Cpp.
22. Середовище CodeBlock.
23. Он-лайн розробка.
24. Засоби відлагодження коду.

Змістовий модуль 2. Мова програмування Сі.

Тема 4. *Мова програмування Сі. Ключові слова, ідентифікатори та модифікатори* (2 год.).

25. Складові частини проекту на Сі.
26. Оператори і операнди.
27. Змінні, оголошення змінних.
28. Ідентифікатори.
29. Ключові слова.
30. Типи даних.
31. Модифікатори типів.
32. Модифікатори доступу.
33. Локальні і глобальні змінні, область видимості змінної.
34. Специфікатори зберігання.

Тема 5. *Оператори* (4 год.).

35. Оператор присвоювання та ініціалізація змінних.
36. Перетворення типів при присвоюванні.
37. Константи, символічні константи з зворотним слешем.
38. Арифметичні оператори.
39. Оператори відношення та логічні оператори.
40. Побітові оператори.
41. Оператор «?».
42. Оператори «&» та «*».
43. Оператор «sizeof».
44. Оператор « , ».
45. Оператори « . » та « -> ».
46. Оператори «[]» та «()».
47. Пріоритети операторів.
48. Приведення типів компілятором.
49. Примусове перетворення типів.
50. Пробіли і дужки.
51. Скорочення операторів.

Тема 6. *Оператори галуження та цикли* (2 год.).

52. Поняття галуження за умовою, оператори: «if» і «switch».
53. Цикли: «while», «for» і «do/while».
54. Оператори переходу: «break», «continue».
55. Конструкція «if-else-if».
56. Розв'язок рівняння алгоритмом половинного ділення.

Тема 7. *Функції* (2 год.).

57. Функції, призначення, оголошення.
58. Попереднє оголошення функції.
59. Оголошення функції через прототип.
60. Прототип функції з невідомим числом параметрів та без параметрів.
61. Правила видимості функцій.

62. Аргументи функцій та передача аргументу через значення або вказівник.
63. Рекурсія.
64. Вказівник на функцію.

Тема 8. *Масиви, стрічки та робота з ними* (4 год.).

65. Одномірні масиви.
66. Багатомірні масиви.
67. Передача масивів в функцію.
68. Алгоритми сортування масивів.
69. Стрічки та стрічкові константи.
70. Функції для роботи з стрічками.
71. Масиви стрічок.
72. Індексунання з допомогою вказівників.
73. Динамічні масиви.
74. Ініціалізація розмірного та безрозмірного масиву.

Тема 9. *Структура пам'яті Сі програми. Вказівники* (2 год.).

75. Структура пам'яті Сі програми.
76. Стек.
77. Купа.
78. Структура пам'яті Сі програми.
79. Динамічне виділення пам'яті і вказівники.
80. Вказівники.
81. Вказівники на константи.
82. Вказівники і масиви.
83. Вказівники на символічні масиви.
84. Масиви вказівників.
85. Вказівники на вказівники.
86. Вказівники на функції.

Тема 10. *Структури* (4 год.).

87. Структури.
88. Доступ до полів структури.
89. Присвоєння структур.
90. Масиви структур.
91. Передача полів структури в функцію.
92. Вказівники на структури.
93. Масиви і структури в структурах.
94. Вкладені структури.
95. Бітові поля.
96. Об'єднання.
97. Переліки.
98. Оператор typedef.

Тема 11. *Система вводу / виводу* (2 год.).

99. Система вводу / виводу.
100. Текстові потоки.

101. Двійкові потоки.
102. Файли, робота з файлами в Сі.
103. Консольний ввід/вивід.

Тема 12. *Робота з файловою системою* (2 год.).

104. Функції файлової системи ANSI Сі.
105. Вказівник на файл.
106. Відкриття, закриття та видалення файлу.
107. Читання і запис символів та стрічок у файл.
108. Функції для роботи з файлами.

Тема 13. *Директиви* (2 год.).

109. Директиви препроцесора, директиви `#define` та `#undef`.
110. Директиви умовної компіляції.
111. Директива `#include`.
112. Оператор «`defined`».
113. Наперед визначені макроси Сі.

4. Структура навчальної дисципліни

Назви змістових модулів і тем	Кількість годин											
	денна форма						заочна форма					
	усього	у тому числі					усього	у тому числі				
		л	п	лаб.	інд.	с. р.		л	п	лаб.	інд.	с. р.
1	2	3	4	5	6	7	8	9	10	11	12	13
Змістовий модуль 1. Основи програмування та алгоритмізації												
Тема 1. Вступ до курсу основи програмування		2		2		6						
Тема 2. Абстракція та алгоритми.		2		2		6						
Тема 3. Компіляція та інтерпретація. Середовище розробки.		2		2		8						
Разом за змістовим модулем 1		6		6		20						
Змістовий модуль 2. Мова програмування Сі												
Тема 4. Мова програмування Сі. Ключові слова, ідентифікатори та модифікатори.		2		2		6						
Тема 5. Оператори.		4		2		6						
Тема 6. Оператори галуження та цикли.		2		2		6						
Тема 7. Функції.		2		2		6						
Тема 8. Масиви, стрічки та робота з ними.		4		4		8						
Тема 9. Структура пам'яті Сі-програми. Вказівники.		2		2		8						
Тема 10. Структури.		4		4		8						
Тема 11. Система вводу/виводу.		2		2		6						
Тема 12. Робота з файловою системою.		2		4		6						
Тема 13. Директиви.		2		2		6						
Разом за змістовим модулем 2		26		26		66						
Усього годин		32		32		86						

5. Теми семінарських занять

Семінарських занять в курсі не передбачено.

6. Теми практичних занять

Практичних занять в курсі не передбачено.

7. Теми лабораторних занять

№ з/п	Назва теми	Кількість годин
1	<i>Основи алгоритмізації. Найпростіші алгоритми.</i>	2
2	<i>Графічна мова програмування Scratch.</i>	2
3	<i>Мова програмування Сі, компіляція, функції вводу/виводу.</i>	2
4	<i>Основні типи змінних. Операції та їх пріоритет.</i>	2
5	<i>Підсумкове заняття 1.</i>	2
6	<i>Умовні оператори.</i>	2
7	<i>Цикли.</i>	2
8	<i>Функції.</i>	2
9	<i>Масиви.</i>	2
10	<i>Підсумкове заняття 2.</i>	2
11	<i>Стрічки.</i>	2
12	<i>Вказівники.</i>	4
13	<i>Структури.</i>	2
14	<i>Робота з файлами.</i>	2
15	<i>Підсумкове заняття 3.</i>	2
	Разом	32

8. Самостійна робота

№ з/п	Назва теми	Кількість годин
1	<i>Вступ до курсу основи програмування</i>	6
2	<i>Абстракція та алгоритми.</i>	6
3	<i>Компіляція та інтерпретація. Середовище розробки.</i>	8
4	<i>Мова програмування Сі. Ключові слова, ідентифікатори та модифікатори.</i>	6
5	<i>Оператори.</i>	6
6	<i>Оператори галуження та цикли.</i>	6
7	<i>Функції.</i>	6
8	<i>Масиви, стрічки та робота з ними.</i>	8
9	<i>Структура пам'яті Сі-програми. Вказівники.</i>	8
10	<i>Структури.</i>	8
11	<i>Система вводу/виводу.</i>	6
12	<i>Робота з файловою системою.</i>	6
13	<i>Директиви.</i>	6
	Разом	86

9. Індивідуальні завдання

10. Методи навчання

Інформаційні методи (лекція, бесіда, ілюстрація, демонстрація); дедуктивні методи на основі узагальнень; евристичні методи (проблемна лекція); інтерактивні методи (дискусія).

Навчальні заняття проводяться у формі лекцій та лабораторних робіт. Лекція – основна форма проведення навчальних занять, призначена для засвоєння теоретичного матеріалу. Під час лабораторних занять студент під керівництвом викладача набуває практичних навичок застосування основ програмування для створення та програмної реалізації типових алгоритмів в процесі практичного написання програмного коду та його відлагодження. Лабораторні заняття проводяться у навчальному комп'ютерному класі.

Самостійна робота студента є основним засобом оволодіння навчальним матеріалом у час, вільний від обов'язкових навчальних занять. Самостійна робота студентів складається з таких видів робіт:

- підготовка до лекцій та лабораторних робіт,
- оформлення звітів про виконані лабораторні роботи,
- самостійне опрацювання окремих тем навчальної дисципліни,
- підготовка до модульних контрольних робіт.

11. Методи контролю

Поточний контроль здійснюється шляхом проведення усного опитування та написання письмових звітів по виконаних лабораторних роботах. У кінці курсу проводиться екзамен.

12. Розподіл балів, які отримують студенти

Поточне тестування та самостійна робота		Лабораторний практикум	Іспит	Сума
Змістовий модуль 1	Змістовий модуль 2			
7	7	36 (3 бали × 12 робіт)	50	100

Шкала оцінювання: національна та ЄКТС

Оцінка ЄКТС	Сума балів за всі види навчальної діяльності	Оцінка за національною шкалою	
		для екзамену, курсового проекту (роботи), практики	для заліку
A	90 – 100	відмінно	зараховано
B	81-89	добре	
C	71-80		
D	61-70	задовільно	
E	51-60		
FX	21-50	незадовільно з можливістю повторного складання	не зараховано з можливістю повторного складання
F	0-20	незадовільно з обов'язковим повторним вивченням дисципліни	не зараховано з обов'язковим повторним вивченням дисципліни

* кількість балів для оцінок «незадовільно» (FX і F) визначається Вченими радами факультетів (педагогічними радами коледжів).

13. Методичне забезпечення

- 1) Хвищун І.О. Алгоритмізація та програмування: Лекції. Матеріали для самостійної роботи. – Львів: Видавничий центр ЛНУ імені Івана Франка, 2017. –278с.
- 2) Шпак З.Я. Програмування мовою С. Навчальний посібник. – Львів: Оріяна-Нова, 2006. – 432 с.
- 3) Вінник В.Ю. Алгоритмічні мови та основи програмування: мова С. Навчальний посібник. – Житомир: ЖДТУ, 2007. – 328 с.

14. Рекомендована література

Основна

1. ISO/IEC 9899:2018 [ISO/IEC 9899:2018] Information technology — Programming languages — C
2. Брайан В. Керніган, Деніс М. Річі. Мова програмування C. Друге видання, 2015. – 232 с. <http://programming.in.ua/programming/c-language/227-book-programming-c-kernighan.html>
3. Хвищун І.О. Алгоритмізація та програмування: Лекції. Матеріали для самостійної роботи. – Львів: Видавничий центр ЛНУ імені Івана Франка, 2017. –278с.

Допоміжна

4. David M. Harris, Sarah L. Harris. Digital Design and Computer Architecture. Second Edition. Elsevier, Inc. -2013. -690 p.
5. Шпак З.Я. Програмування мовою C. Навчальний посібник. – Львів: Оріяна-Нова, 2006. – 432 с.
6. Вінник В.Ю. Алгоритмічні мови та основи програмування: мова C. Навчальний посібник. – Житомир: ЖДТУ, 2007. – 328 с.
7. Daniel Zingaro. Algorithmic Thinking: A Problem-Based Introduction. No Starch Press. – 2020. – 408 p.
8. Robert Seacord. Effective C: An Introduction to Professional C Programming. No Starch Press. – 2020. – 272 p.
9. Anthony Wallit. Learning C programming: An Informative and In-depth Guide to C Programming. Independently published. – 2022. – 250 p.
10. Jeff Szuhay. Learn C Programming: A beginner's guide to learning the most powerful and general-purpose programming language with ease, 2nd Edition. Packt Publishing. – 2022. – 742 p.

15. Інформаційні ресурси

1. Internet – джерела.
2. Наукова бібліотека Львівського національного університету імені Івана Франка (<https://www.lnulibrary.lviv.ua/to-users-2/paid-services/internet/>).
3. Львівська національна наукова бібліотека України імені Василя Стефаника (<https://www.lsl.lviv.ua/index.php/uk/elektronni-resursy1/>).