

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І
ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ**

Кафедра Інформаційних систем та технологій

“ЗАТВЕРДЖУЮ” Директор ННІ Енергетики, автоматики і енергозбереження Каплун В.В. “ ” 20__ р.
“СХВАЛЕНО” на засіданні кафедри Інформаційних систем та технологій Протокол №10 від “15”травня 2023 р. Завідувач кафедри Швиденко М.З.
“РОЗГЛЯНУТО” Гарант ОП к.т.н., доцент, доцент кафедри електропостачання ім. проф. В.М. Синькова Гарант ОП Макаревич С.С.

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Основи програмування

спеціальність 141 «Електроенергетика, електротехніка та електромеханіка»

освітня програма «ІНЖИНІРИНГ ЕЛЕКТРОЕНЕРГЕТИЧНИХ СИСТЕМ З
ВІДНОВЛЮВАНИМИ ДЖЕРЕЛАМИ»

Факультет (ННІ) ННІ енергетики, автоматики і енергозбереження

Розробник: д.т.н., професор, професор кафедри

Інформаційних систем та технологій В.М. Смолій

(посада, науковий ступінь, вчене звання)

Київ – 2023 р.

1. Опис навчальної дисципліни

Основи програмування

(назва)

Галузь знань, спеціальність, освітня програма, освітній ступінь		
Освітній ступінь	Бакалавр	
Спеціальність	141 «Електроенергетика, електротехніка та електромеханіка»	
Освітня програма	Інжиніринг електроенергетичних систем з відновлюваними джерелами	
Характеристика навчальної дисципліни		
Вид	Обов'язкова	
Загальна кількість годин	180	
Кількість кредитів ECTS	6	
Кількість змістових модулів	2	
Курсовий проект (робота) (за наявності)	нема	
Форма контролю	Екзамен	
Показники навчальної дисципліни для денної та заочної форм навчання		
	денна форма навчання	заочна форма навчання
Курс (рік підготовки)	1	
Семестр	2	
Лекційні заняття	30 год.	год.
Практичні, семінарські заняття	-	год.
Лабораторні заняття	90 год.	год.
Самостійна робота	60 год.	год.
Індивідуальні завдання	-	год.
Кількість тижневих аудиторних годин для денної форми навчання	8 год.	

2. Мета, завдання та компетентності навчальної дисципліни

Мета формування у майбутніх фахівців з інжинірингу електроенергетичних систем з ВДЕ сучасного рівня грамотності з основ програмування, опанування основ ефективної роботи у середовищі розробки програм; знайомство з методами, принципами та сучасними підходами до програмування та набуття практичних навичок створення програмного забезпечення для вирішення загальних інженерних задач у рамках спеціальності 141 «Електроенергетика, електротехніка та електромеханіка».

Завдання полягає в отриманні навичок програмування у сучасних середовищах розробки програм; організації вирішення конкретних інженерних задач по розробці алгоритмічного і програмного забезпечення; аналізі створених алгоритмів і програм на предмет оптимальності за певними вимогами, що в

кінцевому результату сприятиме професійній адаптації в сучасному просторі електроенергетики, електротехніки та електромеханіки.

Набуття компетентностей:

інтегральна компетентність (ІК): Здатність розв'язувати професійно-практичні задачі під час провадження професійної діяльності в сфері електричних мереж та електроенергетичних систем або у процесі навчання, що характеризується невизначеністю умов і вимог.

загальні компетентності (ЗК): **ЗК01.** Здатність до абстрактного мислення, аналізу і синтезу.

фахові (спеціальні) компетентності (ФК): **СК19.** Здатність виконувати загальні інженерні розрахунки із застосуванням сучасного програмного забезпечення.

Програмні результати навчання (ПРН):

ПРН06. Застосовувати прикладне програмне забезпечення, мікроконтролери та мікропроцесорну техніку для вирішення практичних проблем у професійній діяльності.

ПРН18. Вміти самостійно вчитися, опановувати нові знання і вдосконалювати навички роботи з сучасним обладнанням, вимірювальною технікою та прикладним програмним забезпеченням.

ПРН26. Знати особливості застосування сучасного програмного забезпечення з метою розв'язання загальних інженерних задач.

3. Програма та структура навчальної дисципліни для:

- повного терміну денної (заочної) форми навчання;
- скороченого терміну денної (заочної) форми навчання.

Назви змістових модулів і тем	Кількість годин													
	денна форма							Заочна форма						
	тижні	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	14	
Змістовий модуль 1. ЗАГАЛЬНІ ПОЛОЖЕННЯ ПРОГРАМУВАННЯ. БАЗОВІ ПОНЯТТЯ МОВИ C														
Тема 1. ПРОСТІ ЕЛЕМЕНТИ МОВИ. ВСТУП	1	9	1		5		3							
Тема 2. СТРУКТУРА ДАНИХ ТА ВИРАЗИ	2	10	2		5		3							
Тема 3. СТРУКТУРА І ПРИКЛАДИ ПРОГРАМИ	3	11	2		5		4							
Тема 4. КЛАСИ ПАМ'ЯТІ. ЛОГІЧНІ ВИРАЗИ. КЕРУЮЧІ СТРУКТУРИ	4	12	2		6		4							
Тема 5. ЛОГІЧНІ	5	12	2		6		4							

ВИРАЗИ. ВІДНОШЕННЯ, ЛОГІЧНІ ОПЕРАЦІЇ, УМОВНІ ВИРАЗИ													
Тема 6. КЕРУЮЧІ СТРУКТУРИ	6	12	2	6	4								
Тема 7. МАСИВИ ТА ПОКАЖЧИКИ	7	12	2	6	4								
Тема 8. ПОКАЖЧИКИ	8	12	2	6	4								
Разом за змістовим модулем 1	90		15	45	30								
Змістовий модуль 2. ПРОГРАМУВАННЯ МОВОЮ С													
Тема 9. МАСИВИ. ПОНЯТТЯ. ПРИКЛАДИ. ОСОБЛИВОСТІ ПРОГРАМУВАННЯ	9	16	3	9	4								
Тема 10. ФУНКЦІЇ. СТРУКТУРИ	10	12	2	6	4								
Тема 11. СТРУКТУРИ АБО ЗАПИСИ	11	12	2	6	4								
Тема 12. ДИРЕКТИВИ ПРЕПРОЦЕСОРА	12	12	2	6	4								
Тема 13. ФАЙЛИ	13	13	2	6	5								
Тема 14. ФУНКЦІЇ ОБМІНУ З ПОТОКАМИ	14	12	2	6	4								
Тема 15. КЕРУВАННЯ БУФЕРІЗАЦІЄЮ	15	13	2	6	5								
Разом за змістовим модулем 2	90		15	45	30								
Усього годин													
Курсовий проект (робота) з _____ <small>(якщо є в робочому навчальному плані)</small>	Не передбачено		-	-	-	-	-	-	-	-	-	-	-
Усього годин	180		30	90	60								

4. Теми семінарських занять. Навчальним планом не передбачено.

5. Теми практичних занять. Навчальним планом не передбачено.

6. Теми лабораторних занять

№ з/п	Назва теми	Кількість годин
1	Арифметичні основи побудови елементів і вузлів обчислювальних машин і систем	4

2	Алгоритмізація задач	4
3	Лінійні обчислювальні процеси	4
4	Умовні конструкції: оператори розгалуження	4
5	Структура програми, основні типи даних, ввід/вивід	4
6	Циклічні конструкції: оператори циклу	4
7	Розгалуження і цикли (у двох частинах)	4
8	Розгалуження і цикли (у двох частинах)	4
9	Змінні і константи, типи даних, ввід та вивід, оператори	4
10	Оператори та цикли	4
11	Функції та рекурсія	4
12	Вказівники і одновимірні масиви даних	4
13	Вказівники	4
14	Масиви. динамічне виділення пам'яті (у двох частинах)	4
15	Масиви. динамічне виділення пам'яті (у двох частинах)	4
16	Робота з масивами (закріплення навичок роботи з масивами)	4
17	Робота зі строками (рядками)	4
18	Строки (рядки) закріплення матеріалу	4
19	Робота зі строками (закріплення навичок роботи зі строками. частина 2)	4
20	Робота зі структурами	4
21	Робота з файлами	4
22	Використання потокових функцій для роботи з текстовими та бінарними файлами	4

7. Теми самостійної роботи

Неформальна освіта на основі проходження онлайн курсу “ Code Yourself! An Introduction to Programming ” на платформі Coursera

№ з/п	Назва теми	Кількість годин
1	МК1. Визначення стану принтеру у DOS	2
2	МК2. Робота зі строковими масивами	2

8. Зразки контрольних питань, тестів для визначення рівня засвоєння знань студентами.

- В чому полягають основні властивості і особливості позиційних систем числення?
- Що називають основою системи числення? 2^n або $-10 \leq n \leq 10$
- Записати число в нормальній формі.
- Сформулювати правило перекладу цілих чисел з однієї системи числення в іншу методом розподілу на основу.
- Сформулювати правило перекладу дробових чисел з однієї системи числення в іншу методом множення на основу.
- Сформулювати правило перекладу чисел з двійкової системи у вісімкову.
- Як виконати над числами наступні арифметичні операції: $A + B + C$; $A - B$; $C - D$; $E - F$; $B * E$; $C * D$; $F : E$; $A : E$. Операцію віднімання виконувати як складання алгебри, представляючи негативні числа в зворотному коді
- Чим викликано застосування додаткового і зворотного кодів?
- В чому відмінності представлення чисел в природній і нормальній формах? Яка

форма і коли зручніше?

10. Як провести складання чисел. $A = 0.1100 * 10^{-101}$, $B = 0.1001 * 10^{-001}$

11. Що таке *алгоритм*?

12. Перерахувати базові структури алгоритмів.

13. Визначити, що являють собою ланцюг та *розгалуження*.

14. Що таке *цикл* як базова структура алгоритму?

15. Які існують різновиди циклів?

16. Що таке «базові типи даних»?

17. Які існують цілочисельні типи у мові C? Скільки байтів пам'яті займає тип *int*?

18. Назвати дійсні типи мови C? Скільки байтів пам'яті займає тип *float*?

19. Як представляються символічні дані у мові C?

20. Що являє собою тип *void*?

21. Що являє собою програма на мові C? Яка структура C-програми?

22. Що таке «операція» та що таке «операнд»? Яка операція називається унарною, а яка – бінарною?

23. Що являє собою вираз?

24. Які арифметичні операції використовуються у мові C?

25. Чи існує логічний тип у мові C? Які значення використовуються для представлення логічних значень?

26. Які використовують операції порівняння та які – логічні операції?

27. Які операції відносяться до операцій присвоювання?

28. Які групи операцій присвоювання існують у мові C? Навести приклади.

29. Чи є оператор програмною одиницею?

30. Які групи основних керуючих конструкцій являють собою оператори?

31. Що таке «пустий оператор» та для чого він використовується?

32. Які існують оператори простої послідовності?

33. Що таке *умовна конструкція*?

34. Структури умовного оператора.

35. Як працює умовний оператор?

36. Що являє собою структура оператора множинного вибору?

37. Як працює оператор множинного вибору?

38. Яка структура програми на мові C?

39. Навіщо потрібна директива *#include*?

40. Що таке *main()*?

41. Перерахувати скалярні типи даних мови C.

42. Що визначає тип даних?

43. Що таке явне і неявне приведення типів? Як і коли воно використовується?

44. Що таке константа? Знайдіть константи у наведених програмах.

45. Що таке змінна?

46. Як проініціалізувати змінну?

47. Чим відрізняється оператор від операції?

48. Чим відрізняються унарні операції від бінарних?

49. Які операції відносяться до арифметичних? Який пріоритет кожної з них?

50. Який порядок виконання операцій у випадку їх однакового пріоритету?

51. Як виконується операція ділення у випадку цілочисельних операндів та у випадку, коли хоч би один з операндів дійсний?

52. Що таке вираз?

53. Яке значення обчислює операція присвоювання?

54. В якому порядку виконується присвоювання у випадку, якщо у виразі їх декілька?

55. Як і навіщо використовуються додаткові операції присвоювання?

56. Чим відрізняються префіксна форма операції інкремента або декремента від постфіксної?

57. Які функції використовуються для вводу інформації? Назвіть їх відмінні особливості.

58. Які функції використовуються для виводу інформації? Назвіть їх відмінні особливості.

59. Чому функції scanf() та printf() називаються функціями форматного вводу та виводу? Як вони працюють?
60. Чим відрізняється керуюча строка функції scanf() від керуючої строки функції printf()?
61. Що таке специфікатор формату? Навіщо він потрібен?
62. Які параметри вказуються функцією scanf() після керуючої строки? Скільки їх повинно бути?
63. Які наслідки невідповідності типу зчитуваної функцією scanf() змінної специфікатору типу?
64. Які параметри вказуються функції printf() після керуючої строки? Скільки їх повинно бути?
65. Які наслідки невідповідності типу виводимого функцією printf() значення специфікатору типу?
66. Що таке керуючі символи? Навіщо вони потрібні? Наведіть приклади.
67. Що таке *цикл*?
68. Які існують два типи циклів?
69. Структура оператора циклу з параметром.
70. Як працює цикл з параметром?
71. Вказати особливості оператора циклу с параметром у мові C?
72. Які цикли відносяться до ітераційних?
73. Навести структуру оператора циклу з передумовою.
74. Як працює цикл з передумовою?
75. Яку структуру має оператор циклу з постумовою?
76. Як працює цикл з постумовою?
77. Чим відрізняється умовна операція від умовного оператора?
78. Що таке повна і неповна форма умовного оператора?
79. Чи може існувати неповна форма умовної операції?
80. Чи потрібно писати "else", якщо при виконанні умови виконується оператор return?
81. Вирази якого типу можуть визначати умови?
82. Які значення виразів, що визначають умови, вважаються істинними, а які хибними?
83. Які операції відносяться до операцій відношення?
84. Чим відрізняються операції "=" від операції "=="?
85. Які операції відносяться до логічних? Який пріоритет їх виконання?
86. Якою операцією можна замінити операцію "&&" ?
87. Якою операцією можна замінити операцію "||" ?
88. Чому може дорівнювати значення виразу відношення або логічного виразу?
89. Як правильно порівняти на рівність дійсні числа?
90. Як правильно перевірити входження значення у певний діапазон?
91. Як перевірити певне цілочисельне значення на рівність нулю?
92. Як перевірити відмінність цілочисельного значення від нуля?
93. Коли використовується вложення умовних операторів?
94. Як правильно записати вложені умовні оператори?
95. Що являє собою оператор switch? Як їм користуватися?
96. Як записати оператор switch з допомогою умовних операторів?
97. Які види циклів снують? Які оператори циклу існують у мові C?
98. Чим відрізняються цикл з передумовою від циклу з постумовою?
99. Коли необхідно використовувати цикл з передумовою, а коли з постумовою? Наведіть приклади.
100. Які цикли з передумовою існують у мові C?
101. Скільки операторів містять у собі тіло циклу у мові C?
102. Як правильно записати цикл з постумовою на мові C?
103. Як задати нескінчений цикл? Навіщо він потрібен? Як з нього вийти?
104. Яким повинно бути значення виразу, який визначає умову виконання циклу, для завершення циклу?
105. Яким повинно бути значення виразу, який визначає умову виконання циклу, для виконання тіла циклу?
106. До чого призведе неправильне завдання виразу, який визначає умови виконання циклу?

107. Чи може бути відсутнім тіло циклу? Якщо може, то наведіть приклади таких циклів.
108. Чим відрізняється оператор *while* від оператора *if*?
109. Який порядок дій при виконанні циклу *for*?
110. Як організувати арифметичний цикл з допомогою циклу *for*?
111. Запишіть алгоритм, який визначається циклом *for*, з допомогою циклу *while*.
112. Що таке вкладений цикл?
113. Скільки разів у загальній складності виконується тіло вкладеного циклу?
114. Як і коли використовуються оператори *break* то *continue*?
115. Що таке рекурентні обчислення? Коли вони використовуються? Як їх програмувати?
116. Що таке функція?
117. Що таке функція типу *void*?
118. Що таке прототип функції?
119. Чим відрізняється прототип функції від виклику функції?
120. Чим відрізняється описання функції від визначення функції?
121. Критерії відповідності формальних і фактичних параметрів.
122. Відповідність типів формальних і фактичних параметрів.
123. Навіщо потрібен оператор *return*?
124. Чи може у функції бути декілька операторів *return*?
125. Коли необхідно писати оператор *return* у функції типу *void*?
126. Що таке побочний ефект функції?
127. Коли використовують формальні параметри-вказівники? Якими у цьому випадку повинні бути фактичні параметри?
128. Як передати масив у функцію? Як передати у функцію матрицю?
129. Як описати функцію, яка дозволяє працювати одночасно з одновимірними масивами, і з матрицями?
130. Що таке параметр-константа? Коли використовують такі параметри
131. Що таке вказівник на функцію? Як його описати?
132. Як передати функцію у функцію? Що буде формальним параметром, а що фактичним? Як
133. звернутися до функції, яку передано через параметри?
134. Що таке рекурсія? Коли вона використовується? Що таке глибина рекурсії?
135. Як описати функцію зі змінною кількістю параметрів? Як використовувати таку функцію?
136. Які параметри можуть бути у функції *main()*? Як налагоджувати і тестувати програму, яка містить функцію *main()* з параметрами?
137. Чим є змінна у програмі?
138. Що означає унарна операція «&», до яких об'єктів вона може застосовуватись?
139. Яке значення може набувати адреса?
140. Чим є вказівник, що він містить?
141. Для чого використовують вказівники?
142. Що означає унарна операція «*», що є результатом її виконання?
143. На об'єкт якого типу може посилатись вказівник?
144. Що означає вказівник на тип *void*?
145. Що таке *статична структура даних*?
146. Чим характеризуються змінні статичних структур?
147. Що являє собою масив даних?
148. На якому етапі виділяється пам'ять під елементи масиву?
149. Чи змінюється обсяг пам'яті, яка виділяється під масив, під час виконання програми?
150. Як занести у вказівник адресу першого елемента масиву?
151. На що посилається будь-який вказівник?
152. Що є результатом операції *вказівник+i*?
153. Чим відрізняються операції *p++* та **p++*, якщо *p* – вказівник?
154. Чим відрізняються операції *(*p)++* та **(++p)*, якщо *p* – вказівник?
155. Що означає лінійний пошук у масиві даних?
156. Що називається сортуванням інформаційної структури?
157. Які існують ознаки порядку?

158. У чому полягає суть бульбашкового сортування масиву даних?
159. Як виконати сортування масиву простим вибором?
160. Що таке вказівник?
161. Який об'єм пам'яті займає вказівник?
162. Що є значенням змінно-вказівника?
163. Як ініціалізувати вказівник?
164. Що таке NULL?
165. Що таке вказівник на void? Навіщо потрібні такі вказівники?
166. Які операції допустимі при роботі з вказівниками?
167. Чим відрізняється унарна операція "&" від унарної "*" ?
168. Сумісність типів вказівників.
169. Чи можна отримати адресу вказівника?
170. Чи можна вказівнику присвоїти його ж адресу?
171. Чому до вказівника на void неможливо застосувати операцію розіменування?
172. Як працюють операції інкременту і декременту, при застосуванні до вказівників?
173. Який результат операції віднімання, яку застосовують до вказівників одного типу?
174. Який специфікатор типу використовують при виводі адреси на екран з допомогою функції printf()?
175. У чому відмінність записів (*double* *) *a* від (*double*) * *a*, якщо *a* – вказівник на ціле число?
176. У чому відмінність записів **a*++ від (**a*)++, якщо *a* – певний вказівник, відмінний від void*?
177. Як описати вказівник на початок масиву?
178. Як описати вказівник на вказівник?
179. Коли і навіщо може повторно використовуватись операція розіменування?
180. Що таке масив?
181. Що являє собою ім'я масиву?
182. Що являє собою індекс елемента масиву?
183. Як можна звернутися до елемента масиву?
184. Як отримати адресу елемента масиву?
185. Як описати вказівник на початок масиву?
186. Як звернутися до елемента масиву через вказівник?
187. Чи існує зв'язок між індексом елемента і його значенням?
188. Чому дорівнює індекс першого елемента масиву?
189. Як проініціалізувати масив?
190. Коли можна не вказувати кількість елементів масиву при описанні?
191. Якого типу можуть бути елементи масиву?
192. Які операції можна здійснювати над цілим масивом?
193. Чому при обробці масивів використовують цикли?
194. Як поміняти місцями два елементи масиву?
195. Чому дорівнює індекс останнього елемента масиву?
196. Чи виникає помилка при зверненні до елемента масиву, індекс якого більше індексу останнього елемента цього масиву?
197. Чи може існувати масив з одного елемента? Якщо може, то як його описати?
198. Чи є *p* вказівником на масив *a*[], якщо вони оголошені наступним чином: *int a[10], *p=a; ?*
199. Що таке сортування масиву?
200. Що таке двовимірний масив?
201. Як розташований двовимірний масив у пам'яті комп'ютера?
202. Як проініціалізувати двовимірний масив?
203. Чому при оголошенні двовимірного масиву з одночасною ініціалізацією перші скобки можна залишити порожніми, а другі - ні?
204. При звертанні до елемента двовимірного масиву спочатку вказується індекс рядку чи індекс стовбця?
205. Як вивести двовимірний масив таблицею з колонками однакової ширини?
206. Що являє собою ім'я двовимірного масиву?

207. Що являє собою перший індекс елемента двовимірного масиву?
208. Що являє собою другий індекс елемента двовимірного масиву?
209. Що являє собою ім'я строки матриці?
210. Як отримати адресу елемента двовимірного масиву?
211. Як отримати адресу початку двовимірного масиву?
212. Як отримати адресу рядка матриці?
213. Чи можна продивитись усі елементи двовимірного масиву в одному циклі?
214. Як звернутися до певного елемента двовимірного масиву через вказівник на перший елемент матриці?
215. Як звернутися до елемента двовимірного масиву через вказівник на початок масиву?
216. Як описати трьохвимірний масив і як з ним працювати?
217. Якими способами можна запрограмувати матрицю?
218. Чим відрізняється перегляд елементів матриці за стовбцями від перегляду за рядками?
219. Який зв'язок між індексами рядка і стовбця у елементів, які знаходяться на головній діагоналі квадратної матриці?
220. Який зв'язок між індексами строки і стовбця у елементів, які лежать на побочній діагоналі квадратної матриці?
221. Чи можна подивитись всі елементи матриці в одному циклі?
222. В яких областях оперативної пам'яті можуть розташовуватись дані при виконанні програми?
223. Які змінні називають динамічними? Чим вони відрізняються від статичних змінних?
224. З якою метою використовують динамічні змінні?
225. Як звертаються до динамічних змінних?
226. Які бувають вказівники? Як їх описати?
227. Яким чином можна виділити пам'ять для динамічних змінних і звільнити її?
228. Як визначити, чи виділена пам'ять чи ні?
229. Чому неможливо забувати звільняти виділену пам'ять?
230. Чому дорівнює значення вказівника після звільнення області пам'яті, на яку він вказував?
231. Як розмістити в динамічній пам'яті масив?
232. Якими способами можна розмістити в динамічній пам'яті матрицю?
233. Що таке C-строка? Чим вона відрізняється від масиву символів?
234. Що таке нуль -термінатор?
235. Чим відрізняється описання *char *st* от *char st[N]*, де N – декотра константа?
236. Як ініціалізувати строку?
237. Чим відрізняється ввід строки з допомогою функції *scanf()* від вводу з допомогою функції *gets()*?
238. Як знайти довжину строки?
239. Чи можна використовувати операцію присвоювання для завдання значення строки?
240. Що таке конкатенація строк?
241. Які існують функції для роботи зі строками?
242. Що таке структурний тип даних?
243. Які види структур даних існують?
244. Що таке вложені структури?
245. Яким чином можна звернутися до різних полів структури?
246. Як задати масив структур?
247. Відкриття і закриття файлів.
248. Читання з текстового файлу і запис в нього.
249. Запис у текстовий файл.
250. Зчитування з двійкового файлу і запис у нього.

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ			
ОС бакалавр ОП ІЕС	Кафедра Інформаційних систем та технологій 2022-2023 навч. рік	ЕКЗАМЕНАЦІЙНИЙ БІЛЕТ № 1 з дисципліни Основи програмування	Затверджую Зав. кафедри (підпис) Швиденко М.З. від 16.05.2023 р.№10
Екзаменаційні запитання			
1. В чому полягають основні властивості і особливості позиційних систем числення?			
2. Які використовують операції порівняння та які – логічні операції?			
3. Як виконується операція ділення у випадку цілочисельних операндів та у випадку, коли хоч би один з операндів дійсний?			
4. Як працює цикл з постумовою?			
5. Скільки операторів містять у собі тіло циклу у мові С?			
6. Що таке побічний ефект функції?			
7. На що посилається будь-який вказівник?			
8. У чому відмінність записів $*a++$ від $(*a)++$, якщо a – певний вказівник, відмінний від void^* ?			
9. Як розташований двовимірний масив у пам'яті комп'ютера?			
10. Які бувають вказівники? Як їх описати?			
Написати мовою С програми для: переведення десяткового числа у двійкове і навпаки, сортування за зростанням елементів одновимірного масиву, знаходження двох перших літер свого прізвища у рядку (строці), створення структури літній відпочинок.			
Тестові завдання різних типів			
Ресурс тестування на навчальному порталі в блоці Підсумкова атестація			
			В.М. Смолій
(підпис)			
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ			
ОС бакалавр ОП ІЕС	Кафедра Інформаційних систем та технологій 2022-2023 навч. рік	ЕКЗАМЕНАЦІЙНИЙ БІЛЕТ № 2 з дисципліни Основи програмування	Затверджую Зав. кафедри (підпис) Швиденко М.З. від 16.05.2023 р.№10
Екзаменаційні запитання			
1. Що називають основою системи числення? 2^n або $-10 \leq n \leq 10$			
2. Які операції відносяться до операцій присвоювання?			
3. Що таке вираз?			
4. Чим відрізняється умовна операція від умовного оператора?			
5. Як правильно записати цикл з постумовою на мові С?			
6. Коли використовують формальні параметри-вказівники? Якими у цьому випадку повинні бути фактичні параметри?			
7. Що є результатом операції $\text{вказівник}+i$?			
8. Як описати вказівник на початок масиву?			
9. Як проініціалізувати двовимірний масив?			
10. Яким чином можна виділити пам'ять для динамічних змінних і звільнити її?			
Написати мовою С програми для: переведення десяткового числа у двійкове і навпаки, сортування за зростанням елементів одновимірного масиву, знаходження двох перших літер свого прізвища у рядку (строці), створення структури літній відпочинок.			
Тестові завдання різних типів			
Ресурс тестування на навчальному порталі в блоці Підсумкова атестація			
			В.М. Смолій
(підпис)			

9. Методи навчання.

При викладанні навчальної дисципліни використовуються такі методи навчання:

- М1. Лекція (інтерактивна, проблемна)
- М2. Лабораторна робота
- М3. Проблемне навчання
- М4. Проєктне навчання (індивідуальне, малі групи, групове)
- М5. Онлайн навчання

10. **Форми контролю.** У курсі передбачено виконання модульного тестування і модульного завдання на МК1 та МК2, підсумкове тестування і виконання підсумкових завдань, співбесіда і врахування результатів неформальної освіти.

При викладанні навчальної дисципліни використовуються такі методи контролю:

- МК1. Тестування
- МК2. Контрольне завдання
- МК3. Розрахункова робота
- МК4. Методи усного контроль (індивідуальне, фронтальне, групове)
- МК5. Екзамен

11. **Розподіл балів, які отримують студенти.** Оцінювання знань студента відбувається за 100-бальною шкалою і переводиться в національні оцінки згідно з табл. 1 «Положення про екзамени та заліки у НУБіП України» (наказ про уведення в дію від 26.04.2023 р. протокол № 10)

Рейтинг студента, бали	Оцінка національна за результати складання	
	екзаменів	заліків
90-100	Відмінно	Зараховано
74-89	Добре	
60-73	Задовільно	
0-59	Незадовільно	Не зараховано

Для визначення рейтингу студента (слухача) із засвоєння дисципліни $R_{\text{дис}}$ (до 100 балів) одержаний рейтинг з атестації (до 30 балів) додається до рейтингу студента (слухача) з навчальної роботи $R_{\text{нр}}$ (до 70 балів): $R_{\text{дис}} = R_{\text{нр}} + R_{\text{ат}}$.

11. Навчально-методичне забезпечення

1. Навчальний курс з дисципліни «Основи програмування»: <https://elearn.nubip.edu.ua/course/view.php?id=802>
2. Методичні рекомендації до виконання лабораторних робіт з дисципліни «Основи програмування» автора В.М.Смолій.

12. Рекомендовані джерела інформації

Література для вивчення курсу

Базова

1. Пол Іре. Об'єктно-орієнтоване програмування з використанням С++: Пер. з англ. - Київ: НИПФ "ДіаСофт Лтд, 1995.
2. Bjarne Stroustrup The C++ Programming language, Addison Wesley, 1986.
3. Фейсон Т. Об'єктно-орієнтоване програмування на Borland C++ 4.5: Пер. з англ. - Київ: Діалектика, 1996. 544с.
4. Сван Т. Опанування Borland C++ 4.5: Пер. з англ. - Київ: Діалектика, 1996. 544с.

Допоміжна

1. ДСТУ 3008-95. Документація. Звіти у сфері науки і техніки. Структура і правила оформлення.

Інтернет-джерела:

1. Draw io. Free online diagram software for making flowcharts, process diagrams, org charts, UML, ER and network diagrams [Електронний ресурс] - <https://app.diagrams.net>
2. Online compiler and debugger for c/c++ [Електронний ресурс] - <https://www.onlinegdb.com>