МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ І.І. МЕЧНИКОВА
Кафедра математичного забезпечення комп'ютерних систем

"ЗАТВЕРДЖУЮ"
Проректор з науково-педагогічної роботи

_____

"_____" _____ 20___ р.

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

ОК11 «Програмування»
(назва навчальної дисципліни)

Рівень вищої освіти *Перший (бакалаврський)*

Галузь знань *12 – Інформаційні технології*

Спеціальність *123 – Комп'ютерна інженерія*
(код і назва спеціальності)

Освітньо-професійна програма *Комп'ютерна інженерія*
(назва ОПП/ОНП)

ОНУ – 2022

Робоча програма навчальної дисципліни «Програмування». – Одеса: ОНУ, 2022. – 21 21 с.

Розробники:
Антоненко О.С., к.ф.-м.н., доцент кафедри МЗКС
Лісіцина І.М., старший викладач кафедри МЗКС

Робоча програма затверджена на засіданні кафедри математичного забезпечення комп'ютерних систем

Протокол № 1 від. "25" 08 2022р.

Завідувач кафедри _____ (підпис) (Євгеній МАЛАХОВ) (Ім'я ПРІЗВИЩЕ)

Погоджено із гарантом ОПП «Комп'ютерна інженерія»

_____ (підпис) (Людмила ВОЛОЩУК) (Ім'я ПРІЗВИЩЕ)

Схвалено навчально-методичною комісією (НМК) з ІТ спеціальностей факультету МФІТ

Протокол № 2 від. "31" 08 2022р.

Голова НМК _____ (підпис) (Алла РАЧИНСЬКА) (Ім'я ПРІЗВИЩЕ)

Переглянуто та затверджено на засіданні кафедри _____

Протокол № 1 від. "29" 08 2023р.

Завідувач кафедри _____ (підпис) (Є.В. Малахов) (Ім'я ПРІЗВИЩЕ)

Переглянуто та затверджено на засіданні кафедри _____

Протокол № ___ від. "____"_____20__ р.

Завідувач кафедри _____ (підпис) (_____) (Ім'я ПРІЗВИЩЕ)

1. Опис навчальної дисципліни

<table>
<tr><td rowspan="2">Найменування показників</td><td rowspan="2">Галузь знань, напрям підготовки, освітньо-кваліфікаційний рівень</td><td colspan="2">Характеристика навчальної дисципліни</td></tr>
<tr><td>денна форма навчання</td><td>заочна форма навчання</td></tr>
<tr><td rowspan="15">Загальна кількість:

кредитів – 13

годин – 390

змістових модулів – 6

ІНДЗ – індивідуальне навчально-дослідне завдання</td><td rowspan="15">Галузь знань
<u>12 – Інформаційні технології</u>
(шифр і назва)

Спеціальність
<u>123 – Комп’ютерна інженерія</u>
(шифр і назва)

Рівень вищої освіти:
<u>Перший (бакалаврський)</u></td><td colspan="2">Обов’язкова</td></tr>
<tr><td colspan="2">Рік підготовки:</td></tr>
<tr><td>1</td><td></td></tr>
<tr><td colspan="2">Семестр</td></tr>
<tr><td>1, 2</td><td>1, 2</td></tr>
<tr><td colspan="2">Лекції</td></tr>
<tr><td>104 год.</td><td>22 год.</td></tr>
<tr><td colspan="2">Практичні, семінарські</td></tr>
<tr><td></td><td></td></tr>
<tr><td colspan="2">Лабораторні</td></tr>
<tr><td>88 год.</td><td>14 год.</td></tr>
<tr><td colspan="2">Самостійна робота</td></tr>
<tr><td>198 год.</td><td>354 год.</td></tr>
<tr><td colspan="2">Індивідуальні завдання:
індивідуальне навчально-дослідне завдання</td></tr>
<tr><td colspan="2">Вид контролю:
залік(1 семестр)

Форма підсумкового контролю:
іспит (2 семестр)</td></tr>
</table>

2. Мета та завдання навчальної дисципліни

Метою дисципліни є придбання базових знань і практичних навичок розробки та налагодження програм на мовах програмування високого рівня відповідно до сучасних технологій та парадигм структурно-процедурного та об'єктно-орієнтованного програмування, вивчення теоретичних основ алгоритмізації і проектування програм, оволодіння базовими алгоритмами, навичками алгоритмічного мислення, навичками розв'язування простіших задач науково-технічного характеру та змісту, засвоєння методів тестування програм та оформлення відповідних презентацій та звітів про виконану роботу, формування у студентів систематичного, інженерного, самостійного, наукового та творчого підходу до освоєння технологій, методів і засобів виробництва програмного забезпечення.

Предметом вивчення курсу є програмні систем, розроблені в структурно-процедурному та об'єктно-орієнтованому стилі, а також їх складові – типи даних, змінні, константи, функції, класи, об'єкти, поля, методи, тощо.

Завдання:

– набуття знань з основ програмування;

– розвиток практичних навичок з розв'язування навчальних і практичних задач програмування;

– освоєння та використання на практиці сучасних технологій програмування (структурне, модульне, об'єктно-орієнтоване програмування);

– освоєння принципів проектування алгоритмів;

– вивчення теоретичних основ алгоритмізації і проектування програм;

– вивчення та практичне засвоєння студентами об'єктно-орієнтованого способу розробки програмних систем;

– навчання студентів використанню сучасного інструментарію розробки, у тому числі: мов програмування C++ та C#, бібліотеки класів STL, інтегрованого середовища програмування Visual Studio чи аналогічного.

Процес вивчення дисципліни спрямований на формування елементів наступних **компетентностей** (згідно ОПП «Комп'ютерна інженерія» від **2022** р.):

а) загальних:

- *Z7. Вміння виявляти, ставити та вирішувати проблеми.*

б) фахових:

- *P2. Здатність використовувати сучасні методи і мови програмування для розроблення алгоритмічного та програмного забезпечення.*
- *P3. Здатність створювати системне та прикладне програмне забезпечення комп'ютерних систем та мереж;*
- *P11. Здатність оформляти отримані робочі результати у вигляді презентацій, науково-технічних звітів.*

Програмні результати навчання:

- *N1. Знати і розуміти наукові положення, що лежать в основі функціонування комп'ютерних засобів, систем та мереж;*
- *N6. Вміти застосовувати знання для ідентифікації, формулювання і розв'язування технічних задач спеціальності, використовуючи методи, що є найбільш придатними для досягнення поставлених цілей;*
- *N8. Вміти системно мислити та застосовувати творчі здібності до формування нових ідей;*
- *N10. Вміти розробляти програмне забезпечення для вбудованих і розподілених застосувань, мобільних і гібридних систем, розраховувати, експлуатувати, типове для спеціальності обладнання;*
- *N11. Вміти здійснювати пошук інформації в різних джерелах для розв'язання задач комп'ютерної інженерії;*
- *N12. Вміти ефективно працювати як індивідуально, так і у складі команди;*
- *N16. Вміти оцінювати отримані результати та аргументовано захищати прийняті рішення.*

Очікувані результати навчання. У результаті вивчення навчальної дисципліни студент повинен

знати*:*

- *поняття інформації та способи її обробки та використання;*
- *системи числення, принципи представлення у комп'ютері цілих і дійсних чисел, а також символів і рядків.*
- *мови низького і високого рівнів; поняття інтерпретації і компіляції програм;*
- *основні етапи життєвого циклу програми: постановка задачі, проектування, кодування, тестування та налагодження, експлуатація;*
- *основні способи запису алгоритмів;*
- *базові алгоритмічні конструкції: послідовне виконання, розгалуження, цикл;*
- *основні операції та оператори мов C/C++;*
- *основні скалярні типи даних мов C/C++;*
- *структуровані типи даних: контейнерні типи, масиви, рядки, структури, класи, списки, файли;*
- *поняття функції, механізми передачі параметрів у функції;*
- *прийоми створення рекурсивних структур і функцій;*
- *команди передпроцесорної обробки мови C/C++;*
- *модульний принцип розробки програм;*
- *способи виділення пам'яті;*
- *правила визначення класів в мовах C++ та C#;*
- *способи взаємодії об'єктів в програмній системі;*
- *основні принципи об'єктно-орієнтованого підходу до програмування – абстракція, інкапсуляція, успадкування і поліморфізм;*
- *правила побудови АТД;*
- *як будувати класи, які повністю або частково моделюють АТД.*

вміти*:*

- *розробляти алгоритми вирішення обчислювальних задач;*
- *складати лінійні програми мовами С/С++;*
- *складати розгалужені програми мовами С/С++;*
- *складати циклічні програми мовами С/С++;*
- *працювати із скалярними даними: цілими, дійсними, символьними, логічними;*
- *складати програми обробки масивів даних;*
- *використовувати функції мов С/С++ при складанні програм;*
- *використовувати структуровані типи даних: масиви, рядки, структури, класи, списки, файли;*
- *використовувати стандартні функції мов С/С++;*
- *виконувати введення-виведення даних;*
- *реалізовувати багатомодульні програми;*
- *складати алгоритми пошуку та сортування масивів;*
- *користуватися засобами інтегрованого середовища розробки Microsoft Visual Studio, onlineGDB чи аналогічного;*
- *здійснювати функціональну та об'єктну декомпозицію програми відповідно до обраної технології програмування;*
- *застосовувати прийоми об'єктно-орієнтованого програмування на практиці;*
- *кодувати програми на об'єктно-орієнтованих мовах програмування, таких як С++ та С#;*
- *використовувати довідкову систему для отримання докладних відомостей про використовувані функції та класи;*
- *виправляти синтаксичні та семантичні помилки та здійснювати рефакторинг коду (налагоджувати та тестувати програму);*
- *використовувати при написанні програм основні стандартні класи STL;*
- *використовувати при написанні програм засоби обробки виняткових ситуацій і інші засоби підвищення якості і надійності програмного коду;*
- *створювати програмні системи відповідно до принципів об'єктно-орієнтованого програмування.*

3. Зміст навчальної дисципліни

Змістовий модуль 1. Алгоритми та структурні мови. Базові елементи програмування.

Тема 1. Введення в архітектуру комп'ютера. Зберігання і обробка даних в комп'ютері. Системи числення. Двійкова, вісімкова та шістнадцяткова системи. Машинні мови та мови високого рівня. Інтерпретатори і компілятори, їх відмінність. Поняття програми та програмування. Різниця інформатики і програмування.

Література: [1-4, 6, 7]

Тема 2. Парадигми програмування. Життєвий цикл програми. Постановка задачі. Технічне завдання. Проектування. Моделі програмних систем. Алгоритми, їх способи запису (перелік пунктів, блок-схема, псевдокод). Властивості алгоритмів. Поняття мови моделювання UML. Кодування. Тестування та налагодження. Експлуатація.

Література: [1-4, 6]

Тема 3. Коротка історія появи мов програмування С та С++. Основні версії та стандарти. Огляд елементів мов С/С++. Алфавіт та лексеми мов С/С++. Ідентифікатори. Ключові слова. Літерали, змінні та типи. Вбудовані типи мов С/С++ та їх літерали. Присвоєння змінних та перетворення типів. Введення та виведення змінних.

Література: [1-4, 6, 7]

Тема 4. Операції та вирази. Арифметичні операції. Операції порівняння. Логічні операції. Побітові операції. Складені оператори присвоєння. Операції інкременту та декременту. Операція sizeof. Обчислення виразів: пріоритет операторів.

Література: [1-4, 6, 7]

Тема 5. Оператори мов С/С++. Поняття про структурне програмування. Оператори розгалуження. Оператори циклу. Оператори переходу.

Література: [1-5, 7]

Тема 6. Масиви. Контейнери мови С++. Масив мови С. Клас vector. Клас array. Клас set – множина. Клас map – асоціативний масив. Використання циклу for для контейнерів. Алгоритми пошуку і сортування.

Література: [1-4, 7]

Тема 7. Функції. Параметри функції (формальні та фактичні). Прототипи функції. Виклик функції. Передача аргументів за значенням, за посиланням і за вказівником. Функція, що не повертає значень (void-функція). Повернення функцією декількох значень. Передача масивів та контейнерів у функції.

Література: [1-4, 7]

Тема 8. Функції стандартної бібліотеки мов С та С++. Генерація випадкових величин. Препроцесор мов С/С++. Директиви #define та #include.

Література: [1-4]

Змістовий модуль 2. Алгоритми та структурні мови. Просунуті елементи програмування.

Тема 1. Рядковий тип. Рядки мови С. Рядки мови С++. Клас string. Введення та виведення рядків. Методи класу string. Передача рядків у функції.

Література: [1-4, 8]

Тема 2. Двовимірні та багатовимірні масиви. Багатовимірні масиви в стилі мови С. Вектор векторів. Передача багатовимірних масивів функції.

Література: [1-4]

Тема 3. Вказівники, посилання та керування пам'яттю. Вказівники. Способи виділення пам'яті, динамічне виділення пам'яті. Поняття адресниго простору процесу; його поділ на частини (код, постійно існуючі дані, динамічна пам’ять, стекова структура – автоматична пам’ять). Адресна арифметика. Динамічні масиви мови С та вказівники. Посилання мови С++. Поняття про зв’язні списки.

Література: [1-4, 6, 8]

Тема 4. Перевантаження функцій. Умови перевантаження. Рекурсивні функції. Шаблонні функції.

Література: [1-4, 6, 7]

Змістовний модуль 3. Об'єктні технології та мови. Основні поняття ООП.

Тема 1. Технології проектування програм. Сучасні критерії якості програм. Потреба використання об'єктно-орієнтованого програмування. Функціональна та об’єктна декомпозиція програми. Об’єктно-орієнтований підхід у програмуванні. Співвідношення між мовами програмування С, С++ і С#. [1-4, 6]

Тема 2. Еволюція програмування від структур до класів. Поняття класу і об'єкта. Поля й методи класів. Вказівник this. [1-4, 6]

Тема 3. Конструктори та деструктори. Перевантаження методів і функцій. Сигнатура методів. [1-4, 6]

Тема 4. Специфікатори доступу. Інкапсуляція. Властивості мови С#. Генерація і обробка виняткових ситуацій. [1-4, 6]

Тема 5. Автоматичне, статичне та динамічне виділення пам’яті для об’єктів. Використання вказівників та посилань на об'єкти. Передача об’єктів у функції. [1-4, 6]

Змістовний модуль 4. Об'єктні технології та мови. Реалізація основних властивостей ООП.

Тема 1. Діаграми класів UML. Відношення композиції об'єктів. [1-4, 6]

Тема 2. Успадкування. Захищені змінні. Управління механізмом доступу до членів базового класу. Особливості використання конструкторів і деструкторів при реалізації механізму успадкування. Виклик затінених методів і базового конструктора. [1-4, 6]

Тема 3. Поліморфізм. Використання покажчиків на похідні типи як основа динамічного поліморфізму. Віртуальні методи класів. [1-4]

Тема 4. Абстрактні класи. Інтерфейси. Механізми успадкування декількох базових класів. [1-4]

Змістовний модуль 5. Об'єктні технології та мови. Об’єктно-орієнтовані технології.

Тема 1. Механізми перевизначення операторів з використанням функцій-членів класу. Поняття про функції-"друзі" класу. Механізми перевизначення операторів з використанням функцій-не членів класу. [1-4]

Тема 2. Особливості реалізації оператора присвоєння. Механізми створення та використання конструктора копії. Використання explicit-конструкторів. Особливості механізму передачі об'єктів функціям і повернення об'єктів функціями. Механізми перевизначення оператора індексації елементів масиву. [1-4]

Тема 3. Організація C++-системи введення-виведення. Поняття "потоків" у мові програмування C++. Особливості механізмів перевизначення операторів введення-виведення даних. Організація файлового введення-виведення даних. [1-4]

Тема 4. Поняття про простори імен. Поняття про статичні члени-даних класу. Статичні методи класу у мові C#. Застосування до функцій-членів класу модифікаторів const і mutable. [1-4]

Змістовний модуль 6. Об'єктні технології та мови. Універсалізація.

Тема 1. Особливості оброблення виняткових ситуацій. Класи виняткових ситуацій. Використання декількох catch-настанов. [1-4]

Тема 2. Побудова АТД та реалізація АТД у вигляді класів. Використання тверджень (assert). [1-4]

Тема 3. Узагальнені (шаблонні) функції та класи. Приклад створення узагальненого класу для організації безпечного масиву. Стандартна бібліотека шаблонів (STL). [1-4]

Для студентів заочної форми на лекції виносяться стисле викладення змістовних модулів 1, 2, 3, 4, 5.

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	ср		л	п	лаб	ср
1	2	3	4	5	6	7	8	9	10	11
Змістовий модуль 1. Алгоритми та структурні мови. Базові елементи програмування.										
Тема 1	12	2		4	6	15	1		0,5	13,5
Тема 2	14	4		4	6	15	1		0,5	13,5
Тема 3	16	6		8	12	15	1		1	27
Тема 4	16	6				15	1			
Тема 5	16	6		8	12	15	1		1	27
Тема 6	14	4				15	1			
Тема 7	14	4		4	6	15	1		0,5	13,5
Тема 8	14	4		4	6	15	1		0,5	13,5
Всього годин 1 семестру	120	36		36	48	120	8		4	108
Змістовий модуль 2. Алгоритми та структурні мови. Просунуті елементи програмування										
Тема 1	26	8		6	12	25	2		1	22
Тема 2	24	8		4	12	25	2		1	22
Тема 3	26	10		4	12	25	2		1	22
Тема 4	24	8		4	12	25	2		1	22
Всього годин по частині «Програмування (Алгоритми та структурні мови)» без ІНДЗ	**220**	**70**		**54**	**96**	220	16		8	196
ІНДЗ	20				20					20
Всього годин по частині «Програмування (Алгоритми та структурні мови)»	**240**	**70**		**54**	**116**	240	16		8	216
Змістовний модуль 3. Об'єктні технології та мови. Основні поняття ООП										
Тема 1.	7	2		2	3	22	1		1	20
Тема 2.	7	2		2	3	22	1		1	20
Тема 3.	7	2		6	10	8	1		2	19
Тема 4.	7	2				7				
Тема 5.	8	2				7				
Змістовний модуль 4. Об'єктні технології та мови. Реалізація основних властивостей ООП										
Тема 1.	8	2		4	8	9	1		1	15
Тема 2.	8	2				8				
Тема 3.	8	2		4	8	9	1		1	15
Тема 4.	8	2				8				
Змістовний модуль 5. Об'єктні технології та мови. Об’єктно-орієнтовані технології										
Тема 1.	8	2		8	15	15	0,5			29
Тема 2.	8	2								
Тема 3.	8	2				15	0,5			
Тема 4.	7	2								
Змістовний модуль 6. Об'єктні технології та мови. Універсалізація										
Тема 1.	9	2		8	15					

Тема 2.	11	3								
Тема 3.	11	3								
Всього годин по частині «Програмування (Об'єктні технології та мови)» без ІНДЗ	**130**	**34**		**34**	**62**	**130**	**6**		**6**	**118**
ІНДЗ	20				20					**20**
Всього годин по частині «Програмування (Об'єктні технології та мови)»	**150**	**34**		**34**	**82**	**150**	**6**		**6**	**138**
Всього годин	**390**	**104**		**88**	**198**	**390**	**22**		**14**	**354**

5. Теми семінарських занять

Семінарські заняття не передбачені

6. Теми практичних занять

Практичні заняття не передбачені

7. Теми лабораторних занять

№ з/п	**Назва теми**	**Кількість годин**
1.	Базові конструкції мови C/C++. Введення та виведення даних з консолі. Побудова лінійних програм.	4
2.	Програмування процесів з розгалуженням у мові C/C++.	8
3.	Реалізація циклічних алгоритмів. Вкладені цикли.	8
4.	Контейнери. Обробка одновимірних масивів.	8
5.	Реалізація алгоритмів з використанням функцій.	4
6.	Масиви як параметри функцій.	4
	Разом за 1 семестр	**36**
7.	Обробка рядків.	6
8.	Двовимірні масиви.	4
9.	Реалізація рекурсивних функцій.	4
10.	Реалізація шаблонних функцій.	4
	Разом по частині «Програмування (Алгоритми та структурні мови)»	**54**
11.	Створення структур та методів їх обробки.	2
12.	Використання стандартних класів C++ (string) та шаблонного класу vector.	2
13.	Створення власних класів. Створення конструкторів класів, методів, використання інкапсуляції.	6
14.	Створення класів-нащадків та композитних класів.	4
15.	Поліморфізм. Абстрактні класи.	4

16.	Створення складного класу з використанням перевизначення операторів.	8
17.	Розробка АТД та перетворення його на клас.	8
	Разом по частині «Програмування (Об'єктні технології та мови)»	**34**
	Разом	**88**

Методичне забезпечення лабораторних робіт – методичні матеріали у електронному вигляді.

Студенти заочної форми виконують лабораторні роботи 1, 2, 3, 4, 5, 6, 7, 11, 12, 13, 14 у повному обсязі, 15 – у зменшеному обсязі.

8. Самостійна робота

№ з/п	Назва теми	Кількість годин
1.	Тема 1. Лабораторна 1. Базові конструкції мови C/C++. Введення та виведення даних з консолі. Побудова лінійних програм. [1]	6
2.	Тема 2. Лабораторна 2. Програмування процесів з розгалуженням у мові C/C++. [1]	6
3.	Тема 3. Тема 4. Лабораторна 3. Реалізація циклічних алгоритмів. Вкладені цикли. [1]	12
4.	Тема 5. Тема 6. Лабораторна 4. Контейнери. Обробка одновимірних масивів. [1]	12
5.	Тема 7. Лабораторна 5. Реалізація алгоритмів з використанням функцій. [1]	6
6.	Тема 8. Лабораторна 6. Масиви як параметри функцій. [1]	6
	Разом за 1 семестр	48
7.	Тема 1. Лабораторна 7. Обробка рядків. [1]	12
8.	Тема 2. Лабораторна 8. Двовимірні масиви. [1]	12
9.	Тема 3. Лабораторна 9. Реалізація рекурсивних функцій. [1]	12
10.	Тема 4. Лабораторна 10. Реалізація шаблонних функцій. [1]	12
11.	Виконання індивідуального навчально-дослідного завдання [2]	20
	Разом по частині «Програмування (Алгоритми та структурні мови)»	116
12.	Тема 1. Лабораторна 11. Створення структур та методів їх обробки. [1]	3
13.	Тема 2. Лабораторна 12. Використання стандартних класів C++ (string) та шаблонного класу vector. [1]	3
14.	Тема 3. Тема 4. Тема 5. Лабораторна 13. Створення власних класів. Створення конструкторів класів, методів, використання інкапсуляції. [1]	10
15.	Тема 1. Тема 2. Лабораторна 14. Створення класів-нащадків та композитних класів. [1]	8
16.	Тема 3. Тема 4. Лабораторна 15. Поліморфізм. Абстрактні класи.	8

	[1]	
17.	Тема 1. Тема 2. Тема 3. Тема 4. Лабораторна 16. Створення складного класу з використанням перевизначення операторів. [1]	15
18.	Тема 1. Тема 2. Тема 3. Лабораторна 17. Розробка АТД та перетворення його на клас. [1]	15
19.	Виконання індивідуального навчально-дослідного завдання [2]	20
	Разом по частині «Програмування (Об'єктні технології та мови)»	**82**
	Разом	**198**

До самостійної роботи відноситься:
[1] – підготовка до лекцій та лабораторних занять
[2] – виконання індивідуального навчально-дослідного завдання

8.1. Курсовий проект

Курсовий проект не передбачений

8.2. Індивідуальне навчально-дослідне завдання

Загальна тема «Розробка програмного забезпечення зі створенням ієрархії класів»

Студент повинен реалізувати ієрархію класів, запропоновану викладачем та використати її у вирішенні прикладної задачі, вказаної в завданні до роботи. При цьому потрібно використовувати основні принципи ООП: інкапсуляцію, успадкування, поліморфізм. Перелік класів задається у варіанті завдання. Студент повинен реалізувати конструктори класів, поля та методи класу. Поля класу мають бути закритими, для доступу до них мають бути описані методи класів. Студенту задається ієрархія класів щодо відношень успадкування та композиції. Де потрібно, використовувати віртуальні методи, чисті віртуальні методи та абстрактні класи для реалізації ідеї динамічного поліморфізму. Використовувати, де необхідно, вбудовані класи мови C++, особливо клас string та шаблонний клас vector. При некоректних вхідних даних методів, конструкторів, функцій генерувати виняткові ситуації, та обробляти їх за допомогою блоків try..catch у функціях більш високого рівня.

Теми:

1. Створення ієрархії класів на тему «*Геометричні фігури: планіметрія*» (а,б)
2. Створення ієрархії класів на тему «*Геометричні фігури: багатогранники*» (а,б)
3. Створення ієрархії класів на тему «*Функції однієї змінної. Поліноми*»
4. Створення ієрархії класів на тему «*Телефонний довідник*»
5. Створення ієрархії класів на тему «*Залізничний вокзал*» (а,б)
6. Створення ієрархії класів на тему «*Стрільба по мішенях*» (а,б)
7. Створення ієрархії класів на тему «*Гра у боулінг*»
8. Створення ієрархії класів на тему «*Музей*» (а,б)
9. Створення ієрархії класів на тему «*Порт*» (а,б,в)
10. Створення ієрархії класів на тему «Обробка тексту з використанням списку слів» (декілька варіантів обробки тексту та різні типи списків - однозв’язні чи двозв’язні)
11. Створення ієрархії класів на тему «*Зоопарк*»
12. Створення ієрархії класів на тему «*Школа*» (а,б)

13. Створення ієрархії класів на тему «*ВНЗ*» (а,б,в)
14. Створення ієрархії класів на тему «*Футбол*» (а,б)
15. Створення ієрархії класів на тему «*Шашки*»
16. Створення ієрархії класів на тему «*Транспортні засоби*»
17. Створення ієрархії класів на тему «Покер»
18. Створення ієрархії класів на тему «*Турклуб*»
19. Створення ієрархії класів на тему «Функції однієї змінної. Диференціювання»
20. Створення ієрархії класів на тему «*Продуктовий магазин*»
21. Створення ієрархії класів на тему «Гіпермаркет» (а,б)
22. Створення ієрархії класів на тему «Будзагін»
23. Створення ієрархії класів на тему «Буфет» (а,б)
24. Створення ієрархії класів на тему «Дартс»
25. Створення ієрархії класів на тему «Запчастини»
26. Створення ієрархії класів на тему «Первісне плем'я»

Приклади докладного завдання по конкретним темам:

Варіант 1а. Створення ієрархії класів на тему «Геометричні фігури: планіметрія»
Створити класи: абстрактний клас «Геометрична фігура» та його підкласи: «Коло», «Трикутник», абстрактний «Чотирикутник» та «Ромб», «Прямокутник», «Трапеція». Реалізувати периметр та площу геометричних фігур та перевірку перетину фігур. Реалізувати зсув фігур. Реалізувати клас «Складна фігура», що поєднує в собі деяку кількість фігур, що не перетинаються. Реалізувати виведення інформації о фігурах в текстовому вигляді.
У проекті передбачити можливість додавання нових фігур, формування складних фігур, обчислення периметру, площі фігур, збереження інформації о фігурах в файлі, та введення її з файлу.

Варіант 1б. Створення ієрархії класів на тему «Геометричні фігури: багатокутники»
Створити класи: абстрактний клас «Геометрична фігура» та його підкласи: абстрактний «Багатокутник», «Трикутник», абстрактний «Чотирикутник» та «Ромб», «Прямокутник», «Трапеція», «Правильний багатокутник». Реалізувати периметр та площу геометричних фігур та перевірку перетину фігур. Реалізувати зсув фігур. Реалізувати виведення інформації о фігурах в текстовому вигляді.
У проекті передбачити можливість додавання нових фігур, обчислення периметру, площі фігур, збереження інформації о фігурах в файлі, та введення її з файлу.

9. Методи навчання

Лекції з використанням мультимедійного презентаційного матеріалу.

10. Методи контролю

Протягом першого семестру студенти виконують 1 тестову та 1 практичну контрольні роботи. Протягом другого семестру студенти виконують 2 тестові та 1 практичну контрольні роботи.
При виконанні **тестових контрольних робіт** студент повинен пройти тест, який складається з теоретичних запитань відкритого типу, направлених на оцінювання

знання студентом теорії та практики, стосовних лінійним, розгалуженим та циклічним алгоритмам, а також математичним та логічним виразам (у першому семестрі) та основним елементам ООП (два тести у другому семестрі). При виконанні **практичних контрольних робіт** студент повинен виконати завдання на теми відповідні модулів – а саме, лінійні, розгалужені, циклічні програми, масиви та функції (у першому семестрі) та рядки, двовимірні масиви, вектори у другому семестрі.

Під час **підсумкового контролю** другого семестру студент повинен відповісти на 2 теоретичні запитання з переліку, наведеному у п. 11.1, та виконати 3 практичних завдання, приклади яких наведені у п. 11.2. Інтегральна оцінка з дисципліни в другому семестрі є усередненою між поточною оцінкою та оцінкою за іспит.

10.1. Критерії оцінювання відповідей на підсумковому контролі

Результат підсумкового контролю оцінюється за 100-бальною шкалою. Мінімальна кількість балів, що зараховується як позитивний результат, дорівнює 60. Бали розподіляються наступним чином: 40 балів – теоретична частина та 60 балів – практична.

На підсумковому контролі студент отримує 2 теоретичні запитання (кожне максимально оцінюється у 20 балів) та 3 практичних завдання, (кожне максимально оцінюється у 20 балів). При оцінюванні якості відповідей беруться до уваги наступні положення.

1. Відповідь повинна бути повною і короткою. Вона не повинна містити матеріал, що не відноситься до суті запитання.
2. Твердження повинні бути сформульовані чітко.
3. Відповіді, що мають помилкові твердження, оцінюються виходячи з близькості відповіді до правильної.
4. Пропуски в обґрунтуванні тверджень призводять до зменшення кількості балів.
5. Недоліки та неточності при викладенні матеріалу зменшують кількість балів.
6. Незнання і нерозуміння основної ідеї теоретичного запитання або практичного завдання призводить до зняття до 90% балів.
7. Якщо відповідь на запитання відсутня, то виставляється нуль балів.

11. Питання до контролю

11.1. Перелік теоретичних питань для підсумкового контролю другого семестру

1. Що таке вказівники, навіщо їх використовувати. Вказівники на структури та об’єкти. Зв’язок масивів та вказівників. Адресна арифметика.
2. Автоматичне, статичне та динамічне виділення пам’яті для об’єктів. Використання вказівників та посилань на об'єкти. Передача об’єктів у функції.

3. Двовимірні та багатовимірні масиви. Багатовимірні масиви в стилі мови C. Вектор векторів. Передача багатовимірних масивів у функції.
4. Перевантаження функцій, конструкторів та методів.
5. Шаблоні функції та класи. Навіщо використовувати. Наведіть приклади.
6. Рекурсивні функції. Вихід з рекурсії. Прямий та обратний ход рекурсії. Використання стеку при рекурсії. Рекурсивні структури даних.
7. Поняття класу та об'єкта. Поля (дані) та методи (поведінка) класів. Конструктори (з параметрами та без параметрів). Перевантаження конструкторів та методів. Приклади конструкторів. Оголошення класів. Створення об'єктів.
8. Клас string. Конструктори класу string. Операції (+, +=, =, порівняння). Методи-запити: length, find. Методи-команди: insert, erase.
9. Що таке інкапсуляція? Як і навіщо ми її використовуємо у C++? Наведіть приклад її використання.
10. Що позначають модифікатори public, protected, private у C++? Які з них зазвичай використовуємо для полів, методів, конструкторів?
11. Що таке конструктор? Навіщо він потрібен? Які синтаксичні особливості має оголошення конструктора порівняно зі звичайним методом? Наведіть приклад конструктора.
12. Що таке наслідування? Коли потрібно та коли небажано його використовувати. Наведіть приклади.
13. Опишіть синтаксис типового конструктора дочірнього класу (підкласу). Наведіть приклад.
14. Що таке поліморфізм? Чим віртуальні методи відрізняються від звичайних методів. Наведіть приклади віртуальних методів.
15. Що таке абстрактний клас? Навіщо він потрібен? Чи можна створити екземпляр абстрактного класу і якщо так, то як? Наведіть приклад абстрактного класу.
16. Коли клас, успадкований від абстрактного, не буде абстрактним? Наведіть приклад.
17. Що таке RTTI (інформація про типи під час виконання)? Навіщо потрібне приведення типів dynamic_cast? Чим воно відрізняється від звичайного наведення типів. Наведіть приклад.

11.2. Приклади практичних завдань для підсумкового контролю другого семестру

1. У заданому рядку порахувати кількість символів у верхньому регістрі.
2. У заданому рядку порахувати кількість входжень заданого символу.
3. У заданому рядку перевести символи ‘D’, ‘E’ та ‘F’ у нижній регістр.
4. У заданому рядку замінити перше входження заданого підрядка на інший підрядок.
5. Дано порожній вектор цілих чисел. Додати до вектора цілі числа від 1 до 22 включно. Додати числа від 23 до 100 включно. Видалити останнє число. Роздрукувати вектор.

6. Ввести з клавіатури число X. Додати перший вектор кратні трьом числа з діапазону від 1 до X, а другий вектор парні з цього діапазону. Надрукувати 12 чисел із першого вектора (якщо чисел у векторі менше 12, надрукувати повідомлення). Роздрукувати другий вектор у зворотному порядку.
7. Скласти програму, яка:
   а) вводить елементи матриці розміром 4 x 5;
   б) визначає номери стовпчика та рядка з мінімальним елементом;
   в) змінює додатні елементи на нулі.
8. Скласти програму, яка:
   а) обчислює елементи матриці розміром 12x7 за формулою $a_{i,j} = \ln(i, j) - i + j$;
   б) знаходить середньоарифметичне усіх додатних елементів матриці;
   в) обчислює суми елементів рядків.
9. Нехай N – невід'ємне ціле число. Рекурсивно описати функцію від параметра N, яка знаходить старшу (ліву) цифру числа N.
10. Нехай N – невід'ємне ціле число. Рекурсивно описати функцію від параметра N, яка знаходить найбільшу цифру числа N.
11. Даний масив дійсних чисел A розміру N. Напишіть рекурсивну функцію, яка знаходить суму позитивних елементів масиву.
12. Написати шаблонну функцію вирішення завдання: знайти максимальне із трьох значень a, b, c.
13. Написати шаблонну функцію вирішення завдання: знайти мінімальне із чотирьох значень a, b, c, d.
14. Написати шаблонну функцію вирішення завдання: знайти максимальний елемент одномірного масиву з парними індексами.
15. Написати шаблонну функцію вирішення завдання: знайти мінімальний елемент одномірного масиву з парними індексами.
16. Написати шаблонну функцію вирішення завдання: знайти перший елемент масиву, більший за деяке значення *limit*.
17. Написати шаблонну функцію вирішення завдання: знайти останній елемент масиву, менший за деяке значення *limit*.

12. Розподіл балів, які отримують студенти

<table>
<tr><td colspan="6">Поточне тестування та самостійна робота – 1 семестр</td><td>Сума – 1 семестр</td></tr>
<tr><td colspan="6">Змістовий модуль №1</td><td rowspan="5">100</td></tr>
<tr><td>Т1</td><td>Т2</td><td>Т3, Т4</td><td>Т5, Т6</td><td>Т7</td><td>Т8</td></tr>
<tr><td>10</td><td>10</td><td>10</td><td>15</td><td>10</td><td>10</td></tr>
<tr><td colspan="3">Тестова контрольна робота за змістовим модулем 1 - 15</td><td colspan="3">Практична контрольна робота за змістовим модулем 1 - 20</td></tr>
</table>

Т1, Т2 ... – теми змістових модулів

Поточне тестування та самостійна робота – 2 семестр								
Змістовий модуль №2				Змістовий модуль №3			Змістовий модуль № 4	
Т1	Т2	Т3	Т4	Т1	Т2	Т3, Т4, Т5	Т1, Т2	Т3, Т4
5	5	5	5	5	5	5	5	5

Поточне тестування та самостійна робота – 2 семестр			Підсумковий контроль (іспит)	Інтегрована оцінка – 2 семестр
Змістовий модуль №5	Змістовий модуль №6	Індивідуальне самостійне завдання	100	100
Т1, Т2, Т3, Т4	Т1, Т2, Т3	25		
15	15			

Т1, Т2 ... – теми змістових модулів

Шкала оцінювання: національна та ECTS

Загальна сума балів	Оцінка ECTS	Національна шкала	
90 — 100	A – «відмінно»	5 «відмінно»	«залік»
85 — 89	B – «дуже добре»	4 «добре»	
75 — 84	C – «добре»		
70 — 74	D – «задовільно»	3 «задовільно»	
60 — 69	E – «допустимо»		
35 — 59	F – «незадовільно з можливістю повторного складання»	2 «незадовільно»	«незалік»
0 — 34	FX – «незадовільно з обов’язковим повторним курсом»		

13. Методичне забезпечення

Комплекс навчально-методичного забезпечення дисципліни; нормативні документи; презентаційні матеріали.

14. Рекомендована література

14.1. Основна література

1. Stanley B. Lippman, Josée Lajoie, Barbara E. Moo, C++ Primer - 5th Edition / Addison-Wesley Professional, 2012. – 1399 p.
2. Stephen Prata, C++ Primer Plus , Sixth Edition / Addison-Wesley Professional, 2011. – 1440 p. ISBN: 0321776402.
3. Bjarne Stroustrup, Programming: Principles and Practice Using C++ 2nd Edition / Addison-Wesley Professional, 2014. – 1312 p.
4. Paul Deitel, C++20 for Programmers: An Objects-Natural Approach (Deitel Developer Series) 3rd Edition / Pearson, 2022. – 960 p.

5. Bjarne Stroustrup, Tour of C++, A (C++ In-Depth Series) 3rd Edition / Addison-Wesley Professional, 2022. – 320 p.
6. Грицюк Ю. І., Рак Т. Є. Об'єктно-орієнтоване програмування мовою C++: навчальний. посібник. Львів: Вид-во ЛДУ БЖД, 2011. 404 с.
7. Бєлов Ю.А. Вступ до програмування мовою C++. / Ю.А. Бєлов, Т.О. Карнаух, Ю.В. Коваль, А.Б. Ставровський. – К.: Видавничо-поліграфічний центр «Київський університет», 2012. – 175 с.
8. Вступ до програмування мовою C++. Організація даних / Т. О. Карнаух, Ю. В. Коваль, М. В. Потієнко, А. Б. Ставровський. – К.: ВПЦ "Київський університет", 2015.
9. Татарчук Д. Д., Діденко Ю. В. Програмування мовами C та C++: навч. посіб. / Д.Д. Татарчук, Ю.В. Діденко. – К.: , 2012. – 112 с.

14.2. Допоміжна література

10. Об'єктно-орієнтоване програмування: конспект лекцій для студентів напряму підготовки "Комп'ютерні науки" всіх форм навчання / Ю. Е. Парфьонов, В. М. Федорченко, М. Ю. Лосєв, О. В. Щербаков. – Харків : Вид. ХНЕУ, 2010. – 312 с.
11. Bertrand Meyer, Object-Oriented Software Construction, second edition, Prentice Hall, 1997. — 1296 p.
12. Шпак З. Я. Програмування мовою C. – Львівська політехніка, 2011. – 436 с. ISBN 978-617-607-104-4
13. Роберт Мартін. Чистий код. Створення і рефакторинг за допомогою Agile. – Фабула, 2019. – 448 с. ISBN: 9786170952851
14. Роберт Мартін. Чиста архітектура. – Фабула, 2020. – 368 с. ISBN: 9786170952868

15. Електронні інформаційні ресурси

1. The C++ Resources Network – Режим доступу: www.cplusplus.com/
2. C++ reference – Режим доступу: https://en.cppreference.com/w/
3. C++ Language Reference / Microsoft Docs – Режим доступу: https://docs.microsoft. com/en-us/cpp/cpp/cpp-language-reference
4. International Standard ISO/IEC 14882:2020(E) – Programming Language C++: [Електронний ресурс]. – Режим доступу: https://isocpp.org/std/the-standard.
5. Презентації лекцій та методичні вказівки до виконання лабораторних робіт в електронному вигляді.
6. Лабораторні роботи на платформі onlineGDB.