

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**МИКОЛАЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**  
**ІМЕНІ В. О. СУХОМЛИНСЬКОГО**  
Механіко-математичний факультет

Кафедра комп'ютерних наук та прикладної математики




**РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ**  
**ПРОГРАМУВАННЯ**

Ступінь бакалавра

Галузь знань 12 Інформаційні технології  
спеціальність 122 «Комп'ютерні науки»  
освітня програма Комп'ютерні науки

2019 – 2020 навчальний рік

Розробник: Ходякова Галина Вікторівна, доцент кафедри комп'ютерних наук та прикладної математики, кандидат педагогічних наук  (Ходякова Г.В.)

Робоча програма затверджена на засіданні кафедри комп'ютерних наук та прикладної математики

Протокол № 1 від «27» серпня 2019 р.

Завідувач кафедри  (Поздєєв В.О.)

«27» серпня 2019 р.

## 1. Опис навчальної дисципліни

Найменування показників	Галузь знань, освітній ступінь	Характеристика навчальної дисципліни	
		<i>денна форма навчання</i>	
Кількість кредитів – 10	Галузь знань 12 Інформаційні технології	Вибіркова	
	Спеціальність 122 Комп'ютерні науки		
Загальна кількість годин 300	Освітня програма: Комп'ютерні науки	<b><i>Рік підготовки:</i></b>	
		3-й	
		<b><i>Семестр</i></b>	
		5-й	
Тижневих годин для денної форми навчання: аудиторних – 4 самостійної роботи студента – 12	Ступінь бакалавра	<b><i>Лекції</i></b>	
		20 год.	
		<b><i>Практичні, семінарські</i></b>	
		<b><i>Лабораторні</i></b>	
		36 год.	
		<b><i>Самостійна робота</i></b>	
244 год.			
		Вид контролю: іспит	

Мова навчання – українська.

### **Примітка.**

Співвідношення кількості годин аудиторних занять до самостійної та індивідуальної роботи становить: для денної форми навчання – 56 год. – аудиторні заняття, 244 год. – самостійна робота (25% ~ 75%).

## 2. Мета, завдання навчальної дисципліни та результати навчання

**Мета** курсу «Програмування» є ознайомлення студентів із основами ООП, поняттями про його базові концепції (інкапсуляція, поліморфізм, наслідування), застосування в подальшому мов програмування, що підтримують методологію ООП (C++, Java, C#). Формування у студентів знань, вмінь і навичок, необхідних для ефективного використання сучасних технологій програмування для розв'язку різних задач та створення прикладного програмного забезпечення загального та професійного спрямування у своїй майбутній професійній, практичній діяльності. У підсумку, вивчення даної дисципліни студент отримує основу для подальшого вивчення усього циклу «комп'ютерних» дисциплін та самостійного оволодіння будь-якими мовами програмування, в основі яких лежить методологія ООП.

**Завдання** – набуття знань та навичок у сфері розробки програмного забезпечення

- вивчити основні принципи об'єктно-орієнтованої парадигми програмування, як найбільш поширеної і затребуваної в даний час;
- вивчити основні можливості об'єктно-орієнтованої мови програмування C++;
- вивчити основні методи програмування на мові C++;
- отримати навички практичного програмування на мові C++;
- отримати навички візуального програмування.

**Студенти повинні знати:**

- основні можливості об'єктно-орієнтованої мови і засоби їх використання;
- будову і загальні підходи до використання засобів стандартних бібліотек;
- основні принципи реалізації класів;
- технології об'єктно-орієнтованої розробки програм;
- технології розробки програм на мові C++;

**вміти:**

- описувати класи, їх атрибути і методи;
- розуміти призначення та використовувати конструктори, деструктори, перевантажені (overloaded) функції та оператори;
- описувати віртуальні (virtual) методи та абстрактні (abstract) класи;
- описувати і використовувати статичні (static) атрибути та методи класів;
- описувати та використовувати дружні (friend) функції класів;
- застосовувати при розробці класів інкапсуляцію, поліморфізм та наслідування (в т.ч. множинне);
- вміти самостійно опановувати нові методи та технології розробки програм.

**Передумови для вивчення дисципліни:** алгоритми та структури даних, дискретна математика, математична логіка, теорія графів, алгебра.

Навчальна дисципліна складається з 10 кредитів.

**Очікувані результати навчання:**

Застосовувати знання основних форм і законів абстрактно-логічного мислення, основ методології наукового пізнання, форм і методів вилучення, аналізу, обробки та синтезу інформації в предметній області комп'ютерних наук.

Використовувати методологію системного аналізу об'єктів, процесів і систем для задач аналізу, прогнозування, управління та проектування процесів у прикладних задачах.

## **I. Загальнопредметні:**

ЗК1. Здатність до абстрактного мислення, аналізу та синтезу.

ЗК17 Професійне володіння комп'ютером та інформаційними технологіями

КЗН-4 Базові знання з системних та кібернетичних наук, необхідних для засвоєння загально-професійних дисциплін з інформатики.

- КЗН-5 Базові знання в галузі інформатики й сучасних інформаційних технологій.
- КІ-5 Знання методів та правил роботи з комп'ютером та роботи в Інтернеті.
- КІ-6 Знання законів, методів та методик проведення наукових та прикладних досліджень.
- КЗП-4 Знання вимог чинних державних та міжнародних стандартів, методів і засобів проектування комп'ютеризованих систем, життєвого циклу їх програмного забезпечення.
- КЗП-5 Знання та розуміння основ програмування, мов різних рівнів та їхніх переваг для розв'язання конкретних задач, методів розроблення програмного забезпечення комп'ютеризованих систем з використанням сучасних технологій.

## **II. Фахові:**

- КСП-7 Знання методів виявлення, формулювання, специфікації, аналізу та трасування вимог до комп'ютеризованих систем на етапі їх проектування, методів проектування та верифікації абстрактної архітектури комп'ютеризованих систем.
- КСП-15 Знання операційних систем (Windows, Unix тощо), системного програмного забезпечення, найбільш розповсюджених пакетів прикладних програм, інформаційних порталів Інтернет, програмних методів захисту інформації в комп'ютеризованих системах та мережах
- КСП-16 Знання базових та спеціалізованих технологій розроблення програмного забезпечення комп'ютеризованих систем.
- КСП-17 Знання методів, методик контролю та тестування правильності роботи програмного забезпечення комп'ютеризованих систем.

### **3. Програма навчальної дисципліни**

#### **Кредит 1. Класифікація мов програмування. Мова C++**

##### **Тема 1.** Класифікація мов програмування. Особливості мови C++

Класифікація мов програмування. Структура програми. Головна функція. Оголошення функції користувача. Формальні та фактичні параметри. Глобальні і локальні змінні. Опис функцій користувача. Виклик функції. Передача параметрів у функції. Основні поняття та термінологія, структур даних, вимоги до структур даних. Масиви і їх передача у функції.

##### **Тема 2.** Класифікація структур даних.

За складністю – прості та інтегровані. За способом представлення – фізичні та логічні. За наявністю зв'язків – зв'язні та незв'язні. За мінливістю – статичні, напівстатичні та динамічні. За впорядкуванням – лінійні і нелінійні. Структури даних для оперативної пам'яті та для зовнішньої пам'яті.

#### **Кредит 2. Алгоритми пошуку і сортування**

##### **Тема 3.** Алгоритми пошуку.

Алгоритм лінійного пошуку. Алгоритм двійкового пошуку. Алгоритми пошуку у текстах.

##### **Тема 4.** Алгоритми сортування.

Прямі методи сортування. Покращені алгоритми сортування. Удосконалені алгоритми сортування.

#### **Кредит 3. Опрацювання текстових даних та файлів**

##### **Тема 5.** Обробка текстових даних. Клас String

Обробка рядків: пошук, заміна, перестановка в рядках. Генерування тексту. Алгоритми пошуку в тексті. Специфіка комп'ютерної обробки літературної мови. Основи обробки текстів у прикладних областях: інформаційно-пошукові системи, електронні словники, переклад та ін.

##### **Тема 6.** Обробка файлів

Поняття файлу. Потоки даних. Виняткові ситуації, пов'язані з роботою з файлами. Основні алгоритми над файловими структурами. Текстові файли та їх обробка. Обробка файлів стандартизованих форматів.

#### **Кредит 4. Стандартна бібліотека STL**

##### **Тема 7.** Стандартна бібліотека шаблонів STL

Структура бібліотеки. Контейнери. Ітератори. Посилання. Типова організація класу контейнера. Розподільники, предикати і функтори.

##### **Тема 8.** Елементарні структури struct

Опис елементарних структур, операції із структурами. Програма-меню. Опрацювання структур.

## **Кредит 5. Основні принципи ООП. Класи та об'єкти.**

**Тема 9.** Огляд об'єктно-орієнтованого програмування.

Історія програмування. Модель об'єктно-орієнтованого програмування. Абстракція даних. Інкапсуляція. Наслідування. Поліморфізм. Методологія ООП.

**Тема 10.** Класи.

Синтаксис оголошення класу. Приклад оголошення класу. Визначення функції-члена класу. Створення об'єкта класу. Використання вказівників на об'єкт. Динамічні об'єкти. Використання операції розширення видимості. Обмеження на дані члени-класу. Неповне оголошення класу. Специфікатори доступу. Створення класу в середовищі розробки Code Blocks.

## **Кредит 6. Елементи класів.**

**Тема 11.** Функції-члени класу.

Конструктори і деструктори. Правила для конструкторів. Правила для деструкторів. Список ініціалізації елементів. Конструктори по замовчуванню. Конструктори копіювання. Вказівник this. Вбудовані функції (inline). Статичні дані-члени класу. Статичні функції-члени класу. Константні дані-члени класу. Константні функції-члени класу. Константні об'єкти.

**Тема 12.** Дружні функції. Дружні класи.

Функції дружні одному класу. Функції дружні декільком класам. Функції-члени дружні іншому класу.

## **Кредит 7. Реалізація принципу наслідування у ООП.**

**Тема 13.** Просте наслідування.

Робота конструктора і деструктора. Базовий клас координат. Похідний клас точки. Похідний клас вектора. Використання конструктора з параметрами. Перевизначення і виклик членів базового класу.

**Тема 14.** Множинне наслідування.

Робота конструктора і деструктора. Приклад множинного наслідування. Віртуальні базові класи.

## **Кредит 8. Механізм перевантаження у ООП**

**Тема 15.** Перевантаження функцій.

Призначення перевантаження. Декорування компілятором імен функцій. Перевантаження функцій. Перевантаження конструкторів. Неоднозначність перевантаження.

**Тема 16.** Перевантаження операторів..

Поняття перевантаження операторів. Правила перевантаження операторів. Перевантаження унарних операторів. Перевантаження оператора присвоювання. Перевантаження бінарних операторів. Перевантаження операторів  $X=$ . Перевантаження бінарних операторів, які використовують об'єкти двох класів. Перевантаження операторів інкременту і декременту. Перевантаження оператора індексування. Перевантаження оператора виклику функції. Перетворення типу.

## **Кредит 9. Принцип поліморфізму у ООП**

**Тема 17.** Віртуальні функції.

Раннє і пізнє зв'язування. Динамічний поліморфізм. Віртуальні функції. Віртуальні деструктори.

**Тема 18.** Абстрактні класи і чисто віртуальні функції.

## **Кредит 10. Обробка виключень.**

**Тема 19.** Потоківне введення – виведення.

Зумовлені потоки. Переваги потоків бібліотеки C++. Операції поміщення і витягання із потоку. Форматування потоку. Файлове введення – виведення з використанням потоків.

**Тема 20.** Обробка виключень.

Обробка виключних ситуацій. Генерування виключень.



#### 4. Структура навчальної дисципліни

Назви змістових модулів і тем	Кількість годин					
	Денна форма					
	Усього	у тому числі				
л		п	лаб	інд	ср	
<b>Кредит 1. Класифікація мов програмування. Мова C++</b>						
<b>Тема 1.</b> Класифікація мов програмування. Мова C++. Структура програми.	14	2		2		10
<b>Тема 2.</b> Класифікація структур даних. Функції користувача	16			2		14
Усього	30	2		4		24
<b>Кредит 2. Алгоритми пошуку і сортування</b>						
<b>Тема 3.</b> Алгоритми пошуку	12	2				10
<b>Тема 4.</b> Алгоритми сортування	18			2		16
Усього	30	2		2		26
<b>Кредит 3. Опрацювання текстових даних та файлів</b>						
<b>Тема 5.</b> Обробка текстових даних Клас String	14	2		2		10
<b>Тема 6.</b> Обробка файлів	16			2		14
Усього	30	2		4		24
<b>Кредит 4. Стандартна бібліотека STL</b>						
<b>Тема 7.</b> Стандартна бібліотека шаблонів STL	16	2		2		12
<b>Тема 8.</b> Елементарні структури struct	14			2		12
Усього	30	2		4		24
<b>Кредит 5. Основні принципи ООП. Класи та об'єкти.</b>						
<b>Тема 9.</b> Огляд об'єктно-орієнтованого програмування.	12	2				10
<b>Тема 10.</b> Класи.	18			2		16

Усього	30	2		2		26
<b>Кредит 6. Елементи класів</b>						
<b>Тема 11.</b> Функції-члени класу.	14	2		2		10
<b>Тема 12.</b> Дружні функції. Дружні класи.	16			2		14
Усього		2		4		24
<b>Кредит 7. Реалізація принципу наслідування у ООП.</b>						
<b>Тема 13.</b> Просте наслідування.	16	2		2		12
<b>Тема 14.</b> Множинне наслідування.	14			2		12
Усього	30	2		4		24
<b>Кредит 8. Механізм перевантаження у ООП</b>						
<b>Тема 15.</b> Перевантаження функцій.	18	2		2		14
<b>Тема 16.</b> Перевантаження операторів	12			2		10
Усього	30	2		4		24
<b>Кредит 9. Принцип поліморфізму у ООП</b>						
<b>Тема 17.</b> Віртуальні функції	14	2		2		10
<b>Тема 18.</b> Абстрактні класи і чисто віртуальні функції.	16			2		14
Усього	30	2		4		24
<b>Кредит 10. Обробка виключень</b>						
<b>Тема 19.</b> Потокowe введення – виведення.	14	2		2		10
<b>Тема 20.</b> Обробка виключних ситуацій. Генерування виключень.	16			2		14
Усього	30	2		4		24
<b>Усього годин:</b>	<b>300</b>	<b>20</b>		<b>36</b>		<b>244</b>

## 5. Теми лекційних занять

№ з/п	Назва теми	Кількість годин
	<i>Кредит 1. Класифікація мов програмування. Мова С++</i>	
1	Тема 1. Класифікація мов програмування. Мова С++. Структура програми.	2
	<i>Кредит 2. Алгоритми пошуку і сортування</i>	
2	Тема 2. Опрацювання масивів у функціях. Пряме сортування масивів. Удосконалене сортування масивів	2
	<i>Кредит 3. Опрацювання текстових даних та файлів</i>	
3	Тема 3. Опрацювання текстових даних класом String. Підключення та робота з файлами.	2
	<i>Кредит 4. Стандартна бібліотека STL</i>	
4	Тема 4. Застосування стандартних структур даних бібліотеки STL	2
	<i>Кредит 5. Основні принципи ООП. Класи та об'єкти.</i>	
5	Тема 5. Створення проектів зі структурами. Створення класу.	2
	<i>Кредит 6. Елементи класів.</i>	
6	Тема 6. Класи та дані-члени класів. Функції-члени класів. Види конструкторів.	2
	<i>Кредит 7. Реалізація принципу наслідування у ООП.</i>	
7	Тема 7. Розробка класів та їх структурних елементів. Дружні функції. Просте наслідування. Множинне наслідування.	2
	<i>Кредит 8. Механізм перевантаження у ООП</i>	
8	Тема 8. Перевантаження унарних та бінарних операторів. Оператори та файли виведення і вставки в потік.	2
	<i>Кредит 9. Принцип поліморфізму у ООП</i>	
9	Тема 9. Віртуальні функції. Абстрактні класи і чисто віртуальні функції.	2
	<i>Кредит 10. Обробка виключень.</i>	
10	Тема 10. Поточе введення – виведення. Обробка виключень.	2
	Разом:	20

## 6. Теми лабораторних занять

№ з/п	Назва теми	Кількість годин
	<i>Кредит 1. Класифікація мов програмування. Мова C++</i>	
1	Тема 1. Передача параметрів у функції.	2
2	Тема 2. Одновимірні та двовимірні динамічні масиви у функціях.	2
	<i>Кредит 2. Алгоритми пошуку і сортування</i>	
3	Тема 3. Опрацювання масивів у функціях	2
4	Тема 4. Пряме та удосконалене сортування масивів	2
	<i>Кредит 3. Опрацювання текстових даних та файлів</i>	
5	Тема 5. Опрацювання текстових даних класом String	2
6	Тема 6. Підключення та робота з файлами	2
	<i>Кредит 4. Стандартна бібліотека STL</i>	
7	Тема 7. Застосування стандартних структур даних бібліотеки STL	2
	<i>Кредит 5. Основні принципи ООП. Класи та об'єкти.</i>	
8	Тема 8. Створення проектів зі структурами.	2
9	Тема 9. Створення класу.	2
	<i>Кредит 6. Елементи класів.</i>	
10	Тема 10. Класи та дані-члени класів. Функції-члени класів	2
11	Тема 11. Види конструкторів	2
	<i>Кредит 7. Реалізація принципу наслідування у ООП.</i>	
12	Тема 12. Розробка класів та їх структурних елементів	2
13	Тема 13. Просте наслідування. Множинне наслідування.	2
	<i>Кредит 8. Механізм перевантаження у ООП</i>	
14	Тема 14. Перевантаження унарних та бінарних операторів.	2
	<i>Кредит 9. Принцип поліморфізму у ООП</i>	
15	Тема 15. Віртуальні функції	2

16	Тема 16. Абстрактні класи і чисто віртуальні функції.	2
	<i>Кредит 10. Обробка виключень.</i>	
17	Тема 17. Віртуальні функції	2
18	Тема 18. Потокове введення – виведення. Обробка виключень.	2
	Разом:	36

### 7. Самостійна робота

№ з/п	Назва теми	Кількість годин
<i>Кредити 1, 2. Класифікація мов програмування. Мова C++. Основні принципи ООП.</i>		
1.	Тема 1. Опрацювання структур	10
2.	Тема 2. Види конструкторів	10
	Оформлення звітів. Підготовка до захисту робіт	24
<i>Кредит 3, 4. Класи та об'єкти. Елементи класів.</i>		
3.	Тема 3. Перевантаження операцій.	10
4.	Тема 4. Абстрактні класи.	10
	Оформлення звітів. Підготовка до захисту робіт	20
<i>Кредит 5. Реалізація принципу наслідування у ООП.</i>		
5.	Тема 5. Шаблони класів.	10
6.	Тема 6. Опрацювання виключних ситуацій.	10
	Оформлення звітів. Підготовка до захисту робіт	20
<i>Кредит 6. Механізм перевантаження</i>		
7.	Тема 7. Перевантаження операцій.	10
8.	Тема 8. Оформлення звітів. Підготовка до захисту робіт	20
<i>Кредит 7, 8. Механізм перевантаження у ООП</i>		
9.	Тема 9. Шаблони класів динамічних масивів.	10
10.	Тема 10. Потокові класи.	10

	Оформлення звітів. Підготовка до захисту робіт	20
<i>Кредит 9, 10. Принцип поліморфізму та обробка виключень.</i>		
11.	Тема 11. Спеціальні класи, функції і вказівники.	10
12.	Тема 12. Препроцесор і компілятор.	10
	Оформлення звітів. Підготовка до захисту робіт	30
Всього :		244

## 8. Форми роботи та критерії оцінювання

Рейтинговий контроль знань студентів здійснюється за 100-бальною шкалою.

Шкала оцінювання: національна та ECTS

ОЦІНКА ЄКТС	СУМА БАЛІВ	ОЦІНКА ЗА НАЦІОНАЛЬНОЮ ШКАЛОЮ	
		екзамен	залік
A	90-100	5 (відмінно)	5/відм./зараховано
B	80-89	4 (добре)	4/добре/ зараховано
C	65-79		
D	55-64	3 (задовільно)	3/задов./ зараховано
E	50-54	2 (незадовільно)	Не зараховано
FX	35-49		

Форми поточного та підсумкового контролю. Комплексна діагностика знань, умінь і навичок студентів із дисципліни здійснюється на основі результатів проведення поточного й підсумкового контролю знань (КР). Поточне оцінювання (індивідуальне, групове і фронтальне опитування, самостійна робота, самоконтроль). Завданням поточного контролю є систематична перевірка розуміння та засвоєння програмового матеріалу, виконання практичних, лабораторних робіт, уміння самостійно опрацьовувати тексти, складання конспекту рекомендованої літератури, написання і захист реферату, здатності публічно чи письмово представляти певний матеріал.

Завданням підсумкового контролю (КР, залік) є перевірка глибини засвоєння студентом програмового матеріалу модуля.

Критерії оцінювання відповідей на практичних заняттях:

Студенту виставляється відмінно, якщо студент здатний самостійно здійснювати основні види навчальної діяльності. Знання студента є глибокими, міцними, узагальненими; студент вміє застосовувати знання творчо, його навчальна діяльність позначена вмінням самостійно оцінювати різноманітні життєві ситуації, явища, факти, виявляти і відстоювати особисту позицію.

Студенту виставляється дуже добре, якщо студент знає істотні ознаки понять, явищ, закономірностей, зв'язків між ними, а також самостійно застосовує знання в нестандартних

ситуаціях, володіє розумовими операціями, вміє робити висновки, виправляти допущені помилки. Відповідь повна, правильна, логічна, обґрунтована.

Студенту виставляється добре, якщо студент знає ознаки понять, явищ, закономірностей, зв'язків між ними на середньому рівні, а також самостійно застосовує знання в стандартних ситуаціях, володіє розумовими операціями, вміє робити висновки, виправляти допущені помилки. Відповідь повна, правильна, логічна, обґрунтована.

Студенту виставляється достатньо, якщо відповідь студента при відтворенні навчального матеріалу елементарна, зумовлюється початковими уявленнями про предмет вивчення. Студент відтворює основний навчальний матеріал, здатний виконувати завдання за зразком, володіє елементарними вміннями навчальної діяльності.

Студенту виставляється мінімальний задовільно, якщо відповідь студента при відтворенні навчального матеріалу елементарна, зумовлюється початковими уявленнями про предмет вивчення. Студент відтворює основний навчальний матеріал.

Кількість балів у кінці семестру повинна складати від 250 до 500 балів (за 5 кредитів), тобто сума балів за виконання усіх завдань.

Відповідний розподіл балів, які отримують студенти за 10 крд

Поточне оцінювання та самостійна робота																				Конт р робот	Нак. бали	
T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	T15	T16	T17	T18	T19	T20			
90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	100	1000

## 9. Засоби діагностики

Засобами діагностики та методами демонстрування результатів навчання є: завдання до лабораторних занять, завдання для самостійної та індивідуальної роботи, презентації результатів досліджень, тестові завдання, контрольні роботи.

## 10. Методи навчання

Курс складається з лекційних, лабораторних занять та самостійної роботи студентів, домашніх завдань і завершується підсумковим рейтинг-контролем і виконанням конкретних контрольних завдань по даній дисципліні.

Лекційні заняття призначені для теоретичного осмислення і узагальнення базових розділів курсу.

Лабораторні заняття є аудиторними, проводяться по наперед відомих темах у вигляді практичної роботи студентів за комп'ютерами. Лабораторне заняття – форма навчального заняття, спрямована на закріплення та вдосконалення студентом теоретичних знань, отриманих як на лекційних заняттях, так і в процесі самостійного вивчення матеріалу. Під час лабораторного заняття студенти під керівництвом викладача особисто набувають практичних навичок у роботі з обчислювальною технікою, оволодівають методикою створення програмних продуктів у програмному середовищі. У ході заняття студенти самостійно виконують передбачені завданням дії, заносючи результати в звіт. На це відводиться до 85 – 90% часу заняття. Викладач здійснює контроль за роботою і надає допомогу при виникненні труднощів, звертає увагу на складні, ключові моменти. Причому основна увага приділяється не вказівці на конкретну помилку, а методиці пошуку причин виникнення цих помилок. Складання звіту – це відповідальний етап лабораторного заняття.

При його складанні студенти розвивають навички аналізу, узагальнення і творчого осмислення результатів роботи, а також навички розробки документації. Кожна лабораторна робота передбачає наявність індивідуального завдання і наступного захисту.

Самостійна робота є позааудиторною і призначена для самостійного ознайомлення студента з певними розділами курсу за рекомендованими педагогом матеріалами і підготовки до виконання індивідуальних завдань по курсу.

Поточний рейтинг-контроль проводиться викладачем в процесі проведення всіх видів занять. Проміжний рейтинг-контроль призначений для практичної комплексної оцінки освоєння розділів курсу і здійснюється шляхом підготовки студентами відповідей на поставлені питання.

Кінцевий контроль знань з дисципліни проводиться під час складання іспиту.

## **11. Методичне забезпечення**

Бібліотека університету має достатню кількість підручників та посібників, лабораторних практикумів та іншої літератури для забезпечення повноцінного навчання студентів.

Крім цього на кафедрі наявна бібліотека для користування студентами на електронних носіях.

Конспекти лекцій, інструкції до лабораторних робіт та додаткові матеріали розміщуються на сервері кафедри та у навчально-методичному комплексі дисципліни.

## **12. Рекомендована література Базова**

1. Роберт Лафоре. Объектно-ориентированное программирование в С++ - Питер, 2013 – 928 с.
2. Бондарев В.М.- Программирование на С++ - Харьков: «Компания СМИТ», 2005 – 284 с.
3. Э. Гамма, Р. Хелм, Р. Джонсон, Д. Влассидес: Приемы ООП. Паттерны проектирования – Питер, 2001 – 368 с.
4. В.В. Мухортов, В.Ю. Рылов: Объектно-Ориентированное Программирование, Анализ и Дизайн – Новосибирск, 2006 – 103 с.
5. Тимоти Бадд: Объектно-ориентированное программирование в действии – 2006
6. Астахова И.Ф., Власов С.В. - Язык С++. Учебное пособие – Мн: Новое название, 2003 – 203 с.

## **Допоміжна**

1. Александреску А. - Современное проектирование на С++ - М: Издательский дом «Вильямс», 2002 – 336 с.
2. Белецкий Я. Энциклопедия языка Си. – М.: Мир, 1992.
3. С++. Язык программирования. – М.: “И.В.К.-СОФТ”, 1991.
4. Березин В.Н., Березин И.В. Начальный курс Си и Си++. – М.: Диалог-Мифи, 1996.
5. Иан Грэхем Объектно-ориентированные методы. Принципы и практика = Object-Oriented Methods: Principles & Practice. — 3-е изд. — М.: «Вильямс», 2004. — С. 880. — ISBN 0-201-61913-X
6. Флорес И. Структуры и управление данными. - М.: Финансы и статистика. 1982.
7. Хэнкон Л., Кригер М. Введение в программирование на языке Си. - М.: Сов. радио, 1986.



8. Э. Гамма, Р. Хелм, Р. Джонсон, Д. Влссидес: Приемы ООП. Паттерны проектирования – Питер, 2001 – 368 с.
9. В.В. Мухортов, В.Ю. Рылов: Объектно-Ориентированное Программирование, Анализ и Дизайн – Новосибирск, 2006 – 103 с.
10. Тимоти Бадд: Объектно-ориентированное программирование в действии – 2006
11. Александреску А. - Современное проектирование на С++ - М: Издательский дом «Вильямс», 2002 – 336 с.
12. Астахова И.Ф., Власов С.В. - Язык С++. Учебное пособие – Мн: Новое название, 2003 – 203 с.
13. Хэнкон Л., Кригер М. Введение в программирование на языке Си. - М.: Сов. радио, 1986.
14. Белецкий Я. Энциклопедия языка Си. – М.: Мир, 1992.
15. С++. Язык программирования. – М.: “И.В.К.-СОФТ”, 1991.
16. Березин В.Н., Березин И.В. Начальный курс Си и Си++. – М.: Диалог-Мифи, 1996.
17. Иан Грэхем Объектно-ориентированные методы. Принципы и практика = Object-Oriented Methods: Principles & Practice. — 3-е изд. — М.: «Вильямс», 2004. — С. 880. — ISBN 0-201-61913-X
18. Антони Синтес Освой самостоятельно объектно-ориентированное программирование за 21 день = Sams Teach Yourself Object-Oriented Programming in 21 Days. — М.: «Вильямс», 2002. — С. 672. — ISBN 0-672-32109-2
19. Ашарина, И.В. Основы программирования на языках С и С++ / И.В. Ашарина. - М.: ГЛТ, 2012. - 208 с.
20. Вирт Н. Алгоритмы и структуры данных. - М.: Мир, 1989. - 360 с.
21. Ахо А., Хопкрофт Д., Ульман Д. Структуры данных и алгоритмы. М.: Вильямс, 2000
22. М. Эллис, Б. Страуструп. Справочное руководство по языку С++ с комментариями: Пер. с англ. - Москва: Мир, 1992. 445с.
23. Стенли Б. Липпман. С++ для начинающих: Пер. с англ. 2тт. - Москва: Унитех; Рязань: Гэлион, 1992, 304-345сс.
24. Бруно Бабэ. Просто и ясно о Borland С++: Пер. с англ. - Москва: БИНОМ, 1994. 400с.
25. В. Липский Комбинаторика для программистов.-1988. 200 с.
26. С.Окулов Программирование в алгоритмах.- 2006. 220с.
27. Серджвик Р. Фундаментальные алгоритмы на С.Часть 1-3 СПб: Диасофт, 2003.
28. Кениг, Э. Эффективное программирование на С++. Практическое программирование на примерах. Т. 2 / Э. Кениг, Б.Э. Му. - М.: Вильямс, 2016. - 368 с.
29. МакГрат, М. Программирование на С для начинающих / М. МакГрат. - М.: Эксмо, 2015. - 192 с.
30. 9. Фленов, М.Е. Программирование на С++ глазами хакера. / М.Е. Фленов. - СПб.: ВHV, 2012. - 352 с.
31. Хенкеманс, Д. Программирование на С++ / Д. Хенкеманс, М. Ли. - СПб.: Символ-плюс, 2015. - 416 с.
32. Бочков С.О. Субботин Д.М. Язык программирования С для персонального компьютера. - М.: Радио и связь, 1990.
33. В.В. Подбельский. Язык С++: Учебное пособие. - Москва: Финансы и статистика, 1995. 560с.
34. Г. Шилдт. Самоучитель С++: Пер. с англ. - Санкт-Петербург: ВHV-Санкт-Петербург, 1998. 620с.

### 13. Інформаційні ресурси

1. [http://ru.wikipedia.org/wiki/Объектно-ориентированное\\_программирование](http://ru.wikipedia.org/wiki/Объектно-ориентированное_программирование)
2. <http://www.youtube.com/watch?v=rbCkcBDobCY>
3. <http://www.codenet.ru/progr/vbasic/oop.php>
4. <http://sdb.su/oop/>
5. <http://younglinux.info/oopython.php>

6. <http://msdn.microsoft.com/ru-ru/library/dd460654.aspx>
7. <http://www.weblibrary.biz/c-sharp/princypy>
8. <http://www.myoop.ru/>
9. <http://phpjs.ru/material-62>
10. <http://proger.elitno.net/oop/121-preimushchestva-i-nedostatki-oop-obektno-orientirovannye-yazyki-programmirovaniya>

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
МИКОЛАЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ В. О. СУХОМЛИНСЬКОГО**

Механіко-математичний факультет

Кафедра комп'ютерних наук та прикладної математики



**ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ  
ПРОГРАМУВАННЯ**

Ступінь бакалавра

Галузь знань 12 Інформаційні технології  
спеціальність 122 «Комп'ютерні науки»  
освітня програма Комп'ютерні науки

Миколаїв – 2019

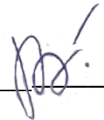
Програму розроблено та внесено: Миколаївський національний університет імені В. О. Сухомлинського

РОЗРОБНИК ПРОГРАМИ: Ходякова Галина Вікторівна, доцент кафедри комп'ютерних наук та прикладної математики, кандидат педагогічних наук.

Програму схвалено на засіданні кафедри комп'ютерних наук та прикладної математики

Протокол від «27» серпня 2019 року № 1

Завідувач кафедри комп'ютерних наук та прикладної математики (Поздєєв В.О.)



Програму погоджено навчально-методичною комісією механіко-математичного факультету

Протокол від «27» серпня 2019 року №

Голова навчально-методичної комісії



(Пархоменко О.Ю.)

Програму погоджено навчально-методичною комісією університету

Протокол від «27» серпня 2019 року № 14

Голова навчально-методичної комісії університету



(Кузнецова О.А.)

Програма вивчення нормативної вибіркової дисципліни «Програмування» складена відповідно до освітнього ступеня підготовки бакалаврів спеціальності 122 Комп'ютерні науки.

**Предметом вивчення** навчальної дисципліни є об'єктно-орієнтоване програмування на C++.

### **1. Мета та завдання навчальної дисципліни**

**Мета** курсу «Програмування» є ознайомлення студентів із основами ООП, поняттями про його базові концепції (інкапсуляція, поліморфізм, наслідування), застосування в подальшому мов програмування, що підтримують методологію ООП (C++, Java, C#). Формування у студентів знань, вмінь і навичок, необхідних для ефективного використання сучасних технологій програмування для розв'язку різних задач та створення прикладного програмного забезпечення загального та професійного спрямування у своїй майбутній професійній, практичній діяльності. У підсумку, вивчення даної дисципліни студент отримує основу для подальшого вивчення усього циклу «комп'ютерних» дисциплін та самостійного оволодіння будь-якими мовами програмування, в основі яких лежить методологія ООП.

**Завдання** – набуття знань та навичок у сфері розробки програмного забезпечення

- вивчити основні принципи об'єктно-орієнтованої парадигми програмування, як найбільш поширеної і затребуваної в даний час;
- вивчити основні можливості об'єктно-орієнтованої мови програмування C++;
- вивчити основні методи програмування на мові C++;
- отримати навички практичного програмування на мові C++;
- отримати навички візуального програмування.

**Студенти повинні знати:**

- основні можливості об'єктно-орієнтованої мови і засоби їх використання;
- будову і загальні підходи до використання засобів стандартних бібліотек;
- основні принципи реалізації класів;
- технології об'єктно-орієнтованої розробки програм;
- технології розробки програм на мові C++;

**вміти:**

- описувати класи, їх атрибути і методи;
- розуміти призначення та використовувати конструктори, деструктори, перевантажені (overloaded) функції та оператори;
- описувати віртуальні (virtual) методи та абстрактні (abstract) класи;
- описувати і використовувати статичні (static) атрибути та методи класів;
- описувати та використовувати дружні (friend) функції класів;
- застосовувати при розробці класів інкапсуляцію, поліморфізм та наслідування (в т.ч. множинне);
- вміти самостійно опановувати нові методи та технології розробки програм.

**Передумови для вивчення дисципліни:** алгоритми та структури даних, дискретна математика, математична логіка, теорія графів, алгебра.

**Очікувані результати навчання:**

Застосовувати знання основних форм і законів абстрактно-логічного мислення, основ методології наукового пізнання, форм і методів вилучення, аналізу, обробки та синтезу інформації в предметній області комп'ютерних наук.

Використовувати методологію системного аналізу об'єктів, процесів і систем для задач аналізу, прогнозування, управління та проектування процесів у прикладних задачах.

### **I. Загальнопредметні:**

ЗК1. Здатність до абстрактного мислення, аналізу та синтезу.

ЗК17 Професійне володіння комп'ютером та інформаційними технологіями

КЗН-4 Базові знання з системних та кібернетичних наук, необхідних для засвоєння загально-професійних дисциплін з інформатики.

- КЗН-5 Базові знання в галузі інформатики й сучасних інформаційних технологій.
- КІ-5 Знання методів та правил роботи з комп'ютером та роботи в Інтернеті.
- КІ-6 Знання законів, методів та методик проведення наукових та прикладних досліджень.
- КЗП-4 Знання вимог чинних державних та міжнародних стандартів, методів і засобів проектування комп'ютеризованих систем, життєвого циклу їх програмного забезпечення.
- КЗП-5 Знання та розуміння основ програмування, мов різних рівнів та їхніх переваг для розв'язання конкретних задач, методів розроблення програмного забезпечення комп'ютеризованих систем з використанням сучасних технологій.

### **II. Фахові:**

- КСП-7 Знання методів виявлення, формулювання, специфікації, аналізу та трасування вимог до комп'ютеризованих систем на етапі їх проектування, методів проектування та верифікації абстрактної архітектури комп'ютеризованих систем.
- КСП-15 Знання операційних систем (Windows, Unix тощо), системного програмного забезпечення, найбільш розповсюджених пакетів прикладних програм, інформаційних порталів Інтернет, програмних методів захисту інформації в комп'ютеризованих системах та мережах
- КСП-16 Знання базових та спеціалізованих технологій розроблення програмного забезпечення комп'ютеризованих систем.
- КСП-17 Знання методів, методик контролю та тестування правильності роботи програмного забезпечення комп'ютеризованих систем.

## **2. Інформаційний обсяг навчальної дисципліни**

### **Кредит 1. Класифікація мов програмування. Мова C++**

#### **Тема 1.** Класифікація мов програмування. Особливості мови C++

Класифікація мов програмування. Структура програми. Головна функція. Оголошення функції користувача. Формальні та фактичні параметри. Глобальні і локальні змінні. Опис функцій користувача. Виклик функції. Передача параметрів у функції. Основні поняття та термінологія, структур даних, вимоги до структур даних. Масиви і їх передача у функції.

#### **Тема 2.** Класифікація структур даних.

За складністю – прості та інтегровані. За способом представлення – фізичні та логічні. За наявністю зв'язків – зв'язні та незв'язні. За мінливістю – статичні, напівстатичні та динамічні. За впорядкуванням – лінійні і нелінійні. Структури даних для оперативної пам'яті та для зовнішньої пам'яті.

### **Кредит 2. Алгоритми пошуку і сортування**

#### **Тема 3.** Алгоритми пошуку.

Алгоритм лінійного пошуку. Алгоритм двійкового пошуку. Алгоритми пошуку у текстах.

#### **Тема 4.** Алгоритми сортування.

Прямі методи сортування. Покращені алгоритми сортування. Удосконалені алгоритми сортування.

### **Кредит 3. Опрацювання текстових даних та файлів**

#### **Тема 5.** Обробка текстових даних. Клас String

Обробка рядків: пошук, заміна, перестановка в рядках. Генерування тексту. Алгоритми пошуку в тексті. Специфіка комп'ютерної обробки літературної мови. Основи обробки текстів у прикладних областях: інформаційно-пошукові системи, електронні словники, переклад та ін.

#### **Тема 6.** Обробка файлів

Поняття файлу. Потоки даних. Виняткові ситуації, пов'язані з роботою з файлами. Основні алгоритми над файловими структурами. Текстові файли та їх обробка. Обробка файлів стандартизованих форматів.

### **Кредит 4. Стандартна бібліотека STL**

#### **Тема 7.** Стандартна бібліотека шаблонів STL

Структура бібліотеки. Контейнери. Ітератори. Посилання. Типова організація класу контейнера. Розподільники, предикати і функтори.

#### **Тема 8.** Елементарні структури struct

Опис елементарних структур, операції із структурами. Програма-меню. Опрацювання структур.

## **Кредит 5. Основні принципи ООП. Класи та об'єкти.**

**Тема 9.** Огляд об'єктно-орієнтованого програмування.

Історія програмування. Модель об'єктно-орієнтованого програмування. Абстракція даних. Інкапсуляція. Наслідування. Поліморфізм. Методологія ООП.

**Тема 10.** Класи.

Синтаксис оголошення класу. Приклад оголошення класу. Визначення функції-члена класу. Створення об'єкта класу. Використання вказівників на об'єкт. Динамічні об'єкти. Використання операції розширення видимості. Обмеження на дані члени-класу. Неповне оголошення класу. Специфікатори доступу. Створення класу в середовищі розробки Code Blocks.

## **Кредит 6. Елементи класів.**

**Тема 11.** Функції-члени класу.

Конструктори і деструктори. Правила для конструкторів. Правила для деструкторів. Список ініціалізації елементів. Конструктори по замовчуванню. Конструктори копіювання. Вказівник this. Вбудовані функції (inline). Статичні дані-члени класу. Статичні функції-члени класу. Константні дані-члени класу. Константні функції-члени класу. Константні об'єкти.

**Тема 12.** Дружні функції. Дружні класи.

Функції дружні одному класу. Функції дружні декільком класам. Функції-члени дружні іншому класу.

## **Кредит 7. Реалізація принципу наслідування у ООП.**

**Тема 13.** Просте наслідування.

Робота конструктора і деструктора. Базовий клас координат. Похідний клас точки. Похідний клас вектора. Використання конструктора з параметрами. Перевизначення і виклик членів базового класу.

**Тема 14.** Множинне наслідування.

Робота конструктора і деструктора. Приклад множинного наслідування. Віртуальні базові класи.

## **Кредит 8. Механізм перевантаження у ООП**

**Тема 15.** Перевантаження функцій.

Призначення перевантаження. Декорування компілятором імен функцій. Перевантаження функцій. Перевантаження конструкторів. Неоднозначність перевантаження.

**Тема 16.** Перевантаження операторів..



Поняття перевантаження операторів. Правила перевантаження операторів. Перевантаження унарних операторів. Перевантаження оператора присвоювання. Перевантаження бінарних операторів. Перевантаження операторів  $X=$ . Перевантаження бінарних операторів, які використовують об'єкти двох класів. Перевантаження операторів інкременту і декременту. Перевантаження оператора індексування. Перевантаження оператора виклику функції. Перетворення типу.

### **Кредит 9. Принцип поліморфізму у ООП**

**Тема 17.** Віртуальні функції.

Раннє і пізнє зв'язування. Динамічний поліморфізм. Віртуальні функції. Віртуальні деструктори.

**Тема 18.** Абстрактні класи і чисто віртуальні функції.

### **Кредит 10. Обробка виключень.**

**Тема 19.** Потокове введення – виведення.

Зумовлені потоки. Переваги потоків бібліотеки C++. Операції поміщення і витягання із потоку. Форматування потоку. Файлове введення – виведення з використанням потоків.

**Тема 20.** Обробка виключень.

Обробка виключних ситуацій. Генерування виключень.

## **3. Рекомендована література**

### **Базова**

1. Роберт Лафоре. Объектно-ориентированное программирование в C++ - Питер, 2013 – 928 с.
2. Бондарев В.М.- Программирование на C++ - Харьков: «Компания СМИТ», 2005 – 284 с.
3. Э. Гамма, Р. Хелм, Р. Джонсон, Д. Влссидес: Приемы ООП. Паттерны проектирования – Питер, 2001 – 368 с.
4. В.В. Мухортов, В.Ю. Рылов: Объектно-Ориентированное Программирование, Анализ и Дизайн – Новосибирск, 2006 – 103 с.
5. Тимоти Бадд: Объектно-ориентированное программирование в действии – 2006
6. Астахова И.Ф., Власов С.В. - Язык C++. Учебное пособие – Мн: Новое название, 2003 – 203 с.

### **Допоміжна**

7. Александреску А. - Современное проектирование на C++ - М: Издательский дом «Вильямс», 2002 – 336 с.
8. Белецкий Я. Энциклопедия языка Си. – М.: Мир, 1992.
9. C++. Язык программирования. – М.: “И.В.К.-СОФТ”, 1991.
10. Березин В.Н., Березин И.В. Начальный курс Си и Си++. – М.: Диалог-Мифи, 1996.

11. Иан Грэхем Объектно-ориентированные методы. Принципы и практика = Object-Oriented Methods: Principles & Practice. — 3-е изд. — М.: «Вильямс», 2004. — С. 880. — ISBN 0-201-61913-X
12. Флорес И. Структуры и управление данными. - М.: Финансы и статистика. 1982.
13. Хэнкон Л., Кригер М. Введение в программирование на языке Си. - М.: Сов. радио, 1986.
14. Э. Гамма, Р. Хелм, Р. Джонсон, Д. Влссидес: Приемы ООП. Паттерны проектирования – Питер, 2001 – 368 с.
15. В.В. Мухортов, В.Ю. Рылов: Объектно-Ориентированное Программирование, Анализ и Дизайн – Новосибирск, 2006 – 103 с.
16. Тимоти Бадд: Объектно-ориентированное программирование в действии – 2006
17. Александреску А. - Современное проектирование на С++ - М: Издательский дом «Вильямс», 2002 – 336 с.
18. Астахова И.Ф., Власов С.В. - Язык С++. Учебное пособие – Мн: Новое название, 2003 – 203 с.
19. Хэнкон Л., Кригер М. Введение в программирование на языке Си. - М.: Сов. радио, 1986.
20. Белецкий Я. Энциклопедия языка Си. – М.: Мир, 1992.
21. С++. Язык программирования. – М.: “И.В.К.-СОФТ”, 1991.
22. Березин В.Н., Березин И.В. Начальный курс Си и Си++. – М.: Диалог-Мифи, 1996.
23. Иан Грэхем Объектно-ориентированные методы. Принципы и практика = Object-Oriented Methods: Principles & Practice. — 3-е изд. — М.: «Вильямс», 2004. — С. 880. — ISBN 0-201-61913-X
24. Антони Синтес Освой самостоятельно объектно-ориентированное программирование за 21 день = Sams Teach Yourself Object-Oriented Programming in 21 Days. — М.: «Вильямс», 2002. — С. 672. — ISBN 0-672-32109-2
25. Ашарина, И.В. Основы программирования на языках С и С++ / И.В. Ашарина. - М.: ГЛТ, 2012. - 208 с.
26. Вирт Н. Алгоритмы и структуры данных. - М.: Мир, 1989. - 360 с.
27. Ахо А., Хопкрофт Д., Ульман Д. Структуры данных и алгоритмы. М.: Вильямс, 2000
28. М. Эллис, Б. Страуструп. Справочное руководство по языку С++ с комментариями: Пер. с англ. - Москва: Мир, 1992. 445с.
29. Стенли Б. Липпман. С++ для начинающих: Пер. с англ. 2тт. - Москва: Унитех; Рязань: Гэлион, 1992, 304-345сс.
30. Бруно Бабэ. Просто и ясно о Borland С++; Пер. с англ. - Москва: БИНОМ, 1994. 400с.
31. В. Липский Комбинаторика для программистов.-1988. 200 с.
32. С.Окулов Программирование в алгоритмах.- 2006. 220с.
33. Серджвик Р. Фундаментальные алгоритмы на С.Часть 1-3 СПб: Диасофт, 2003.
34. Кениг, Э. Эффективное программирование на С++. Практическое программирование на примерах. Т. 2 / Э. Кениг, Б.Э. Му. - М.: Вильямс, 2016. - 368 с.
35. МакГрат, М. Программирование на С для начинающих / М. МакГрат. - М.: Эксмо, 2015. - 192 с.
36. 9. Фленов, М.Е. Программирование на С++ глазами хакера. / М.Е. Фленов. - СПб.: ВHV, 2012. - 352 с.
37. Хенкеманс, Д. Программирование на С++ / Д. Хенкеманс, М. Ли. - СПб.: Символ-плюс, 2015. - 416 с.
38. Бочков С.О. Субботин Д.М. Язык программирования С для персонального компьютера. - М.: Радио и связь, 1990.
39. В.В. Подбельский. Язык С++: Учебное пособие. - Москва: Финансы и статистика, 1995. 560с.
40. Г. Шилдт. Самоучитель С++: Пер. с англ. - Санкт-Петербург: ВHV-Санкт-Петербург, 1998. 620с.

## Інформаційні ресурси

1. [http://ru.wikipedia.org/wiki/Объектно-ориентированное\\_программирование](http://ru.wikipedia.org/wiki/Объектно-ориентированное_программирование)
2. <http://www.youtube.com/watch?v=rbCkcBDobCY>
3. <http://www.codenet.ru/progr/vbasic/ooop.php>
4. <http://sdb.su/ooop/>
5. <http://younglinux.info/ooopython.php>
6. <http://msdn.microsoft.com/ru-ru/library/dd460654.aspx>
7. <http://www.weblibrary.biz/c-sharp/principy>
8. <http://www.myoop.ru/>
9. <http://phpjs.ru/material-62>
10. <http://proger.elitno.net/ooop/121-preimushchestva-i-nedostatki-ooop-obektno-orientirovannye-yazyki-programmirovaniya>

### 4. Форма підсумкового контролю успішності навчання: іспит

#### Засоби діагностики успішності навчання:

Лекційні та лабораторні заняття з використанням традиційних методів, інтерактивних методик та комп'ютерної техніки, індивідуальна робота, самостійна робота.

Поточний та підсумковий контроль здійснюється у вигляді комп'ютерних тестів на освітньому просторі університету. Для оцінювання використовується національна чотирьохбальна шкала: відмінно, добре, задовільно, незадовільно; європейська шкала: А, В, С, D, E, FX, F.

(приклад для заліку) 100% балів студенти накопичують на заняттях та під час поточного і підсумкового контролю, що регламентується робочою програмою викладача. (приклад для іспиту) 60% балів студенти накопичують на заняттях та під час поточного контролю, що регламентується робочою програмою викладача, 40% балів студенти набирають на іспиті.