

## СИЛАБУС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ «ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ»



<b>Ступінь освіти</b>	бакалавр
<b>Спеціальність</b>	014 Середня освіта. Інформатика 113 Прикладна математика 123 Комп'ютерна інженерія 126 Інформаційні системи та технології
<b>Тривалість викладання</b>	2 семестр
<b>Заняття:</b>	3, 4 чверті
Лекції	2 години на тиждень
Лабораторні	2 години на тиждень
<b>Мова викладання</b>	українська

**Кафедра, яка викладає:** Інформаційних технологій та комп'ютерної інженерії

**Сторінка курсу в СДО НТУ «ДП»:** <https://do.nmu.org.ua/course/view.php?id=5459>

**Консультації:** за окремим розкладом, що попередньо погоджений зі здобувачами освіти.

**Онлайн-консультації:** MS Teams, електронна пошта

### Інформація про викладача:



<b>Викладач:</b>	Соколова Наталя Олегівна, к.т.н., доц.
<b>Персональна сторінка:</b>	<a href="https://it.nmu.org.ua/ua/HR_staff/prepods/Sokolova.php">https://it.nmu.org.ua/ua/HR_staff/prepods/Sokolova.php</a>
<b>E-mail:</b>	Sokolova.N.O@nmu.one

### 1. Анотація до курсу

Об'єктно орієнтоване програмування (ООП) – це одна з сучасних парадигм програмування, яка базується на ствердженні того, що програма це сукупність об'єктів які взаємодіють між собою. Кожен об'єкт в цій моделі є незалежним, і він здатний отримувати, обробляти дані та відправляти ці дані іншим об'єктам.

Увага в курсі об'єктно-орієнтованого програмування концентрується на побудові обгрунтованих ієрархій, складанні абстрактних інтерфейсів та універсальних сигнатур методів. Ідеал об'єктно-орієнтованого програмування - «програми-пристосуванці» (programming for reuse), придатні до налагодження та функціонування за нових умов лише завдяки дописуванню, але не переписуванню коду. Сьогодні необхідно вчитися програмувати, заглядаючи у майбутнє, розробляти гнучкі програми, придатні до нових, можливо, навіть несподіваних застосувань. Образно кажучи, парадигму «reuse» можна сформулювати, як спробу відкласти на завтра все, чого можна не уточнювати сьогодні.

Базові аспекти об'єктно-орієнтованого програмування однакові для всіх сучасних мов програмування. Головна мета курсу - оволодіння технікою об'єктно-орієнтованого програмування на основі мови програмування C++. Ця мова багатопарадигменна, тому її

вивчення служить доброю базою для порівняльного аналізу парадигм імперативного програмування. Крім цього, мова C++ є розширенням мови програмування C, яка залишаючись мовою програмування високого рівня, дотримується концепції, що дозволяє низкорівневу роботу з машинними структурами даних (бітами, байтами, машинними словами), вказівниками, машинною арифметикою, програмованим розподілом пам'яті тощо.

## 2. Мета та завдання навчальної дисципліни

**Мета** – формування компетентностей щодо формування системи теоретичних і практичних знань в області об'єктно-орієнтованого програмування, в оволодінні основними принципами, методиками і методами розробки алгоритмів і програм на основі об'єктно-орієнтованого підходу, вміння розробляти класи та застосовувати їх для розв'язку задач на ЕОМ, а також придбання практичних навичок з розробки програмного забезпечення на основі об'єктно-орієнтованого підходу з використанням мови програмування C++ та із застосуванням сучасних інтегрованих середовищ розробки програм.

Реалізація мети вимагає трансформації програмних результатів навчання в дисциплінарні та адекватний відбір змісту навчальної дисципліни за цим критерієм.

Основними **завданнями** вивчення дисципліни є

- розгляд та засвоєння основних положень об'єктно-орієнтованого програмування, визначень та вмісту базових понять "інкапсуляція", "успадкування", "поліморфізм", а також принципів організації взаємодії об'єктів;

- оволодіння методиками об'єктно-орієнтованого мовою програмування C++ з вивченням засобів подання даних простими (базовими) та складними (похідними) типами, формування вміння застосовувати різні алгоритмічні конструкції мови та розробляти відповідні до задач класи, реалізуючи основні концепції ООП: інкапсуляцію, спадкування та поліморфізм;

- освоєння методик об'єктно-орієнтованого програмування в сучасних інтегрованих середовищах розробки програм;

- вивчення особливостей та освоєння спеціальних засобів програмування в операційному середовищі Windows.

## 3. Результати навчання

### Спеціальність 113 Прикладна математика

1. Формалізувати задачі, сформульовані мовою певної предметної галузі; формулювати їх математичну постановку та обирати раціональний метод вирішення; розв'язувати отримані задачі аналітичними та чисельними методами, оцінювати точність та достовірність отриманих результатів.
2. Поєднувати методи математичного та комп'ютерного моделювання з неформальними процедурами експертного аналізу для пошуку оптимальних рішень.
3. Вміти застосовувати сучасні технології програмування та розроблення програмного забезпечення, програмної реалізації чисельних і символьних алгоритмів

### Спеціальність 123 Комп'ютерна інженерія

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

### Спеціальність 126 Інформаційні системи та технології

1. Використовувати базові знання інформатики й сучасних інформаційних систем та технологій, навички програмування, технології безпечної роботи в комп'ютерних мережах, методи створення баз даних та інтернет-ресурсів, технології розроблення

- алгоритмів і комп'ютерних програм мовами високого рівня із застосуванням об'єктно-орієнтованого програмування для розв'язання задач проектування і використання інформаційних систем та технологій.
2. Використовувати сучасні методи програмного забезпечення, яке є інформаційних систем та технологій.
  3. Розробляти комп'ютерні системи розпізнавання та обробки зображень, зокрема аерокосмічних зображень, обробки просторових даних, а також програмне забезпечення різноманітної спрямованості.

#### 4. Структура курсу

##### ЛЕКЦІЇ

###### 1 Вступ. Основні положення

Об'єктно-орієнтований аналіз та проектування.

Поняття об'єктно-орієнтованого аналізу, проектування та програмування.

Об'єктна модель предметного середовища, принципи її побудови.

###### 2 Основи об'єктно-орієнтованої мови програмування

Типи даних. Структури даних. Об'єднання

Класи. Абстрагування даних та інкапсуляція

Статичні, константні члени класів

Перевантаження функцій-членів класа

Масиви об'єктів

Об'єкти як аргументи та результати функцій

Вказівники на об'єкти. Вказівник this

###### 3 Конструктори та деструктори

Конструктори

Перевантаження конструкторів

Конструктор копіювання

Деструктори

Резервування і звільнення пам'яті за допомогою конструктора та деструктора

Динамічне виділення пам'яті під об'єкти

Дружні функції та методи

Дружні класи

Вкладені класи

###### 4 Спадкування

Конструктори та деструктори при спадкуванні

Управління доступом похідних класів

Існуючі неіснуючі члени класа

Просте та множинне спадкування

Ієрархія класів

###### 5 Поліморфізм

Реалізація поліморфізму

Статичний поліморфізм. Перевантаження операцій та функцій

Динамічний поліморфізм. Шаблони функцій і класів

Динамічний поліморфізм Віртуальні методи

Абстрактні класи

Механізм пізнього зв'язування

Параметризовані (шаблонні) функції

Параметризовані (шаблонні) класи

Стандартна бібліотека шаблонів

Обробка виняткових ситуацій

## ЛАБОРАТОРНІ ЗАНЯТТЯ

1. Структури даних
2. Динамічні структури даних
3. Поняття класу. Інкапсуляція.
4. Конструктори та деструктори
5. Поліморфізм. Перевантаження операцій
6. Шаблонні класи
7. Спадкування. Віртуальні функції.

### Технічне обладнання та/або програмне забезпечення

1. Персональний комп'ютер або ноутбук зі сталим доступом до мережі Інтернет
2. Активованій акаунт університетської пошти (student.i.p.@nmu.one) на Офіс365.
3. Активний обліковий запис у системі дистанційної освіти Moodle.
4. Програмне забезпечення:
  - Microsoft Office;
  - Середовище розробки C++.
  -

## 5. Система оцінювання та вимоги

**5.1. Навчальні досягнення** здобувачів вищої освіти за результатами вивчення курсу оцінюватимуться за шкалою, що наведена нижче:

Рейтингова шкала	Інституційна шкала
90 – 100	відмінно
75 – 89	добре
60 – 74	задовільно
0 – 59	незадовільно

5.2. Здобувач вищої освіти може отримати **підсумкову оцінку** з навчальної дисципліни на підставі поточного оцінювання знань за умови, якщо набрана кількість балів з поточного тестування та самостійної роботи складатиме не менше 60 балів.

Поточна успішність складається з оцінок за лекційну частину курсу та практичні заняття. Отримані бали осереднюються і є підсумковою оцінкою за вивчення навчальної дисципліни. Максимально за поточною успішністю здобувач вищої освіти може набрати 100 балів.

Максимальне оцінювання:

Теоретична частина	Практична частина		Разом (середнє)
	При своєчасному складанні	При несвоєчасному складанні	
20	80	70 і нижче	100

Практичні завдання приймаються за контрольними запитаннями до кожної роботи.

Теоретична частина оцінюється за результатами задачі контрольних робіт, яка містить 3 задачі.

### 5.3. Критерії оцінювання теоретичної частини курсу.

Під час проведення контрольних заходів наприкінці третьої, четвертої чверті здобувачі вищої освіти складають контрольну, яка містить 3 задачі. Максимальна оцінка за тест складає 10 балів. Контрольна проводиться з використанням системи дистанційної освіти Офіс365.

#### **5.4. Критерії оцінювання практичних робіт.**

З кожної практичної роботи здобувач вищої освіти оформлює звіт, що містить завдання, результати його виконання та висновки. При своєчасному (протягом тижня від дати проведення заняття) та вірному виконанні завдання здобувач отримує оцінку 100. При несвоєчасному виконанні завдання та/або наявності помилок оцінка складає 90 балів і нижче.

### **6. Політика курсу**

#### **6.1. Політика щодо академічної доброчесності**

Академічна доброчесність здобувачів вищої освіти є важливою умовою для опанування результатами навчання за дисципліною і отримання задовільної оцінки з поточного та підсумкового контролів. Академічна доброчесність базується на засудженні практик списування (виконання письмових робіт із залученням зовнішніх джерел інформації, крім дозволених для використання), плагіату (відтворення опублікованих текстів інших авторів без зазначення авторства), фабрикації (вигадування даних чи фактів, що використовуються в освітньому процесі). Політика щодо академічної доброчесності регламентується положенням "Положення про систему запобігання та виявлення плагіату у Національному технічному університеті "Дніпровська політехніка". [http://www.nmu.org.ua/ua/content/activity/us\\_documents/System\\_of\\_prevention\\_and\\_detection\\_of\\_plagiarism.pdf](http://www.nmu.org.ua/ua/content/activity/us_documents/System_of_prevention_and_detection_of_plagiarism.pdf).

У разі порушення здобувачем вищої освіти академічної доброчесності (списування, плагіат, фабрикація), робота оцінюється незадовільно та має бути виконана повторно. При цьому викладач залишає за собою право змінити тему завдання.

#### **6.2. Комунікаційна політика**

Здобувачі вищої освіти повинні мати активовану університетську пошту.

Усі письмові запитання до викладачів стосовно курсу мають надсилатися на університетську електронну пошту.

#### **6.3. Політика щодо перескладання**

Роботи, які здаються із порушенням термінів без поважних причин оцінюються на нижчу оцінку. Перескладання відбувається із дозволу деканату за наявності поважних причин (наприклад, лікарняний).

#### **6.4. Відвідування занять**

Для здобувачів вищої освіти денної форми відвідування занять є обов'язковим. Поважними причинами для неявки на заняття є хвороба, участь в університетських заходах, академічна мобільність, які необхідно підтверджувати документами. Про відсутність на занятті та причини відсутності здобувач вищої освіти має повідомити викладача або особисто, або через старосту.

За об'єктивних причин (наприклад, міжнародна мобільність) навчання може відбуватись в он-лайн формі за погодженням з керівником курсу.

#### **6.5. Політика щодо оскарження оцінювання**

Якщо здобувач вищої освіти не згоден з оцінюванням його знань він може опротестувати виставлену викладачем оцінку у встановленому порядку.

#### **6.6. Студентоцентризований підхід**

Для врахування інтересів та потреб студентів на початку вивчення курсу здобувачам вищої освіти пропонується відповісти у системі Moodle на низку питань щодо інформаційного наповнення курсу. Відповідно до результатів опитування формується траєкторія навчання з урахуванням потреб студентів.

Під час навчання студенти реалізують своє право вибору індивідуальних завдань практичних завдань.

Наприкінці вивчення курсу та перед початком сесії здобувачам вищої освіти пропонується анонімно заповнити у системі Moodle електронні анкети для оцінки рівня задоволеності методами навчання і викладання та врахування пропозицій стосовно покращення змісту навчальної дисципліни. За результатами опитування вносяться відповідні корективи у робочу програму та силабус.

## 7. Рекомендовані джерела інформації

1. Соколова Н.О., Гнатушенко В.В., Суцевський Д.В. ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ НА C++. Навчальний посібник. Дніпро, ПП «Ліра» 2017, – 150с.
2. Соколова Н.О. ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ Опорний конспект лекцій. Дніпро, ПП «Ліра», 2019.
3. Hari Mohan Pandey. Object-Oriented Programming C++ Simplified. Laxmi Publications Pvt Ltd 2017, 802 P.
4. Siddhartha Rao. C++ in One Hour a Day, 2017. -793 P.
5. Malini Devi. C++ Programming Language for Beginners with Easy tips. 2018 - 1146 P.
6. Mike McGrath. C++ Programming, 5th Edition. 2017, 273 P.
7. Ivor Horton & Peter Van Weert. Beginning C++17 From Novice to Professional. 2018, 788 P.
8. Herbert Schildt. C++: The Complete Reference, Fourth Edition, 2003. 1058p.
9. Bjarne Stroustrup. The C++ Programming Language (4th Edition). 2013, 1376 p.
10. Google C++ . Style Guide, available at <https://google.github.io/styleguide/cppguide.html> (дата звернення: 12.12.2020).
11. The Stanford University C++ Style Guide, available at <https://hownot2code.com/2017/01/18/the-stanford-university-c-style-guide/> (дата звернення: 12.12.2020).