

**Силабус курсу**  
**Мови та технології програмування**

**Ступінь вищої освіти – бакалавр**  
**Спеціальність 015.39 «Професійна освіта. Цифрові технології»**  
**Освітньо-професійна програма «Професійна освіта. Цифрові технології»**

**Рік навчання: 3, Семестр: 5**

**Кількість кредитів: 5 Мова викладання: українська**

**Керівник курсу**

**ППП**

PhD, Семененко Юрій Сергійович

**Контактна  
інформація**

[y.semenenko@wunu.edu.ua](mailto:y.semenenko@wunu.edu.ua), +380973631487

**Опис дисципліни**

Навчальна дисципліна «Мови та технології програмування» є складовою освітньої програми підготовки бакалаврів і спрямована на забезпечення студентів необхідними теоретичними знаннями та практичними навичками для розуміння, розробки та використання сучасних мов програмування та технологій.

Основною метою курсу є вивчення принципів програмування, структур даних і алгоритмів, а також розуміння процесів розробки програмного забезпечення. Студенти ознайомляться з різними парадигмами програмування (об'єктно-орієнтоване, процедурне, функціональне), синтаксисом і семантикою сучасних мов програмування та набудуть навички роботи з інструментами і середовищами розробки.

Курс також охоплює питання безпеки, ефективності та масштабованості програмних рішень. Студенти навчатимуться аналізувати та вирішувати задачі програмування, вибирати відповідні мови та технології для розробки програмних рішень, а також застосовувати методи тестування та відлагодження коду.

## Структура курсу

Години (лек. / сем.)	Тема	Результати навчання	Завдання
2/2	Тема 1. Вступ до мов програмування та їх класифікація	Студенти зрозуміють сутність мови програмування та її важливу роль у процесі розробки програмного забезпечення. Вони зможуть простежити історію розвитку мов програмування, що дозволить їм глибше усвідомити еволюцію інструментів для створення програм. Класифікація мов за рівнями – машинні, асемблерні та високорівневі – допоможе студентам орієнтуватися в різноманітті сучасних мов, а також зрозуміти, як мови для різних типів програмування можуть бути використані для вирішення конкретних завдань.	Тести, питання, виконання практичних завдань
2/2	Тема 2. Основні парадигми програмування. Процедурне, об'єктно-орієнтоване, функціональне.	Студенти дослідять базові принципи процедурного програмування, зокрема поняття функцій та процедур, а також вивчать змінні та їхню область видимості. Оволодіння концепціями об'єктно-орієнтованого програмування допоможе їм працювати з класами, об'єктами та застосовувати наслідування, інкапсуляцію і поліморфізм. Також студенти навчатимуться функціональному підходу, який підкреслює чисті функції, відсутність побічних ефектів та важливість рекурсії.	Тести, питання, виконання практичних завдань
2/2	Тема 3. Алгоритми та структури даних у програмуванні	Після опанування цієї теми студенти зможуть використовувати базові алгоритми для сортування та пошуку даних, а також працювати з такими структурами даних, як масиви, списки, стеки, черги, дерева та графи. Важливим аспектом буде розуміння оцінки складності алгоритмів, що дозволить студентам краще оптимізувати програмні рішення та вибирати ефективні алгоритми для конкретних задач.	Тести, питання, виконання практичних завдань

2/2	Тема 4. Об'єктно-орієнтоване програмування (ООП)	Студенти глибше розберуться в основних принципах об'єктно-орієнтованого програмування та навчатимуться створювати класи й об'єкти, застосовувати наслідування та поліморфізм у різних мовах програмування. Знання багаторівневого наслідування та практичного застосування поліморфізму дозволять їм ефективно розвивати програми з використанням ООП у мовах Python, C++ та Java.	Тести, питання, виконання практичних завдань
2/2	Тема 5. Функціональне програмування, основи та застосування	Студенти досліджуватимуть концепції функцій вищого порядку, лямбда-функцій та каррінгу. Вони зможуть зрозуміти, коли і як використовувати функціональний підхід у реальних задачах, а також оцінити його переваги та обмеження. Ця тема дозволить студентам відкрити нові можливості для створення гнучких та масштабованих рішень.	Тести, питання, виконання практичних завдань
2/2	Тема 6. Синтаксис та семантика сучасних мов програмування	Опанувавши основні синтаксичні правила в таких мовах, як Python, C++ та JavaScript, студенти навчатимуться писати правильний код із врахуванням семантичних особливостей кожної з мов. Вони зможуть розуміти обробку змінних, працювати з різними типами даних та аналізувати код на семантичному рівні для правильного розуміння його функціонування.	Тести, питання, виконання практичних завдань
4/4	Тема 7. Модулі, бібліотеки та фреймворки у програмуванні	Студенти навчатимуться створювати модульні програми, використовуючи основні принципи модульності у програмуванні. Знання популярних бібліотек та фреймворків, таких як Django, Flask, React, Angular, дозволить їм обирати ефективні інструменти для створення веб-додатків та інших програмних рішень	Тести, питання, виконання практичних завдань
4/4	Тема 8. Робота з файлами та базами даних у програмах	Після вивчення цієї теми студенти зможуть працювати з файлами різних форматів та базами даних. Вони навчатимуться зчитувати та записувати дані, використовувати основні SQL-запити, а також підключати бази даних до своїх програм через відповідні бібліотеки.	Тести, питання, виконання практичних завдань

4/4	Тема 9. Принципи безпечного програмування	Опанувавши основи безпеки програмного забезпечення, студенти зможуть захищати свої програми від поширених вразливостей, таких як буферні переповнення, SQL-ін'єкції та міжсайтовий скриптинг (XSS). Вони також отримують базові знання в криптографії та навчаться реалізовувати захист даних у своїх програмних рішеннях.	Тести, питання, виконання практичних завдань
4/4	Тема 10. Розробка програмного забезпечення, життєвий цикл проєкту	Студенти засвоять основні етапи розробки програмного забезпечення, від планування до впровадження. Вивчення моделей життєвого циклу, таких як водоспадна модель та Agile, допоможе їм розуміти процеси управління розробкою. Знання інструментів для контролю версій, таких як Git, і управління проєктами, таких як Jira або Trello, дозволить студентам ефективно співпрацювати в командних проєктах.	Тести, питання, виконання практичних завдань

### РЕКОМЕНДОВАНІ ДЖЕРЕЛА ІНФОРМАЦІЇ

1. Raharjo, Teguh, and Betty Purwandari. "Agile Project Management Challenges and Mapping Solutions: A Systematic Literature Review." Proceedings of the 3rd International Conference on Software Engineering and Information Management. 2020.
2. Данченко, О. Б., Занора, В. О. Проектний менеджмент: управління ризиками та змінами в процесах прийняття управлінських рішень : монографія. Черкаси, 2019. 278 с
3. Яковенко О. І. Управління проєктами та ризиками : навч. посібник. Ніжин : Видавець ПП Лисенко М.М., 2019. 196 с
4. Cicala G. The Project Managers Guide to Microsoft Project 2019. Apress, 2020. 681 p.
5. Dionisio C. S. Microsoft Project 2019 For Dummies John Willey & Sons, Inc., 2019. 352 p
6. Васильєв О. Характеристики Програмування C++ в прикладах і задачах. Навч. пос. Збільшений формат В5 Видавництво Ліра-К., 2020 382 с.
7. Пекарський Б.Г. Основи програмування: Навчальний посібник.- Кондор,2018.-364 с.
8. C++ Crash Course: A Fast-Paced Introduction./ Lospinoso Josh. ISBN 1593278885. - 2019.-792с. 5. International Standard ISO/IEC 14882:2014(E) – Programming Language C++ , ISBN-13: 978-0321563842: [Електронний ресурс]. – Режим доступу: <https://isocpp.org/std/the-standard>.
9. C/C++ language and standard libraries reference: [Електронний ресурс]. – Режим доступу: <https://msdn.microsoft.com/en-us/library/hh875057.aspx>.
10. C++ Tutorial: [Електронний ресурс]. – Режим доступу <https://www.w3schools.com/cpp/>.

### Політика оцінювання

*Політика щодо дедлайнів і перескладання.* Для виконання індивідуальних завдань і проведення контрольних заходів встановлюються конкретні терміни. Перескладання модулів відбувається з дозволу дирекції факультету за наявності поважних причин (наприклад, тимчасова втрата працездатності).

*Політика щодо академічної доброчесності.* Використання друкованих і електронних джерел інформації під час контрольних заходів та екзаменів заборонено.

*Політика щодо відвідування.* Відвідування занять є обов'язковим компонентом оцінювання. За об'єктивних причин (наприклад, карантин, воєнний стан, хвороба, закордонне стажування) навчання може відбуватись в он-лайн формі за погодженням із керівником курсу з дозволу дирекції факультету.

Підсумковий бал (за 100-бальною шкалою) з дисципліни «Моделі та технології програмування» визначається як середньозважена величина, в залежності від питомої ваги кожної складової залікового кредиту:

Модуль 1		Модуль 2		Модуль 3	Модуль 4	Модуль 5
10%	10%	10%	10%	5%	15%	40%
Поточне оцінювання	Модульний контроль 1	Поточне оцінювання	Модульний контроль 2	Тренінг	Самостійна робота	Екзамен
Оцінка визначається середнє арифметичне отриманих оцінок за перший змістовий модуль. (теми 1-7)	Виконання модульного завдання, складається із однієї практичної задачі (теми 1-7)	Оцінка визначається середнє арифметичне отриманих оцінок за змістовий модуль. (теми 7-14)	Виконання модульного завдання, складається із однієї практичної задачі (теми 7-14)	Оцінка за виконання завдання (звіт)	Оцінка за виконання самостійного завдання (презентація або звіт)	2 теоретичні запитання (по 30 балів), 1 практичне завдання (40 балів)

Оцінювання здійснюється шляхом опитування не рідше як один раз на два заняття.

**Шкала оцінювання:**

За шкалою Університет	За національною шкалою	За шкалою ECTS
90-100	Відмінно	A (відмінно)
85-89	Добре	B (дуже добре)
75-84		C (добре)
65-74	Задовільно	D (задовільно)
60-64		E (достатньо)
35-59	Незадовільно	FX (незадовільно з можливістю повторного складання)
1-34		F (незадовільно з обов'язковим повторним курсом)