

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Львівський національний університет імені Івана Франка
Факультет прикладної математики та інформатики
Кафедра теорії оптимальних процесів

Затверджено

На засіданні кафедри теорії оптимальних процесів факультету прикладної математики та інформатики Львівського національного університету імені Івана Франка

Протокол № 1 від 28.08.2024 р.)

Завдувач кафедри Шахно С.М.



Силабус з навчальної дисципліни
“Сучасні технології програмування”,
що викладається в межах ОПП «Системний аналіз» другого
(магістерського) рівня вищої освіти для здобувачів з
спеціальності 124 – системний аналіз

Львів 2024 р.

Назва дисципліни	Сучасні технології програмування
Адреса викладання дисципліни	Головний корпус ЛНУ ім. І. Франка м. Львів, вул. Університетська 1
Факультет та кафедра, за якою закріплена дисципліна	Факультет прикладної математики та інформатики Кафедра теорії оптимальних процесів
Галузь знань, шифр та назва спеціальності	12 – інформаційні технології 124 – системний аналіз
Викладачі дисципліни	Недашковський Микола Олександрович, професор кафедри теорії оптимальних процесів, Голуб Богдан Михайлович, доцент кафедри теорії оптимальних процесів, Шунькін Юрій Валерійович асистент кафедри теорії оптимальних процесів.
Контактна інформація викладачів	mykola.nedashkovskyy@lnu.edu.ua ; https://ami.lnu.edu.ua/employee/nedashkovskyy-mykola-oleksandrovych ; bohdan.holub@lnu.edu.ua ; https://ami.lnu.edu.ua/employee/holub-yuriy.shunkina@lnu.edu.ua Головний корпус ЛНУ ім. І. Франка, каб. 269. м. Львів, вул. Університетська, 1
Консультації з питань навчання по дисципліні відбуваються	Консультації в день проведення лекцій/практичних занять (за попереднього домовленістю).
Сторінка курсу	https://ami.lnu.edu.ua/course/suchasni-tekhnologii-prohramuvannia
Інформація про дисципліну	Дисципліна “Сучасні технології програмування” є нормативною дисципліною зі спеціальності 124 – системний аналіз для освітньої програми Системний аналіз, яка викладається у 1-му семестрі в обсязі- 5х кредитів (за Європейською Кредитно-Трансферною Системою ECTS).
Коротка анотація дисципліни	Курс розроблено таким чином, щоб надати учасникам базові знання сучасних технологій програмування, перелік яких формується на основі рейтингу використання алгоритмічних мов
Мета та цілі дисципліни	Метою вивчення нормативної дисципліни “Сучасні технології програмування” є надання студентам базових знань та практичних навичок використання сучасних технологій програмування.

Література для вивчення дисципліни	<ol style="list-style-type: none"> 1. Основи JavaScript / https://developer.mozilla.org/uk/docs/Learn/Getting_started_with_the_web/JavaScript_basics 2. Java Підручник / https://w3schoolsua.github.io/java/index.html#gsc.tab=0 3. Python Підручник / https://w3schoolsua.github.io/python/index.html#gsc.tab=0 4. PHP Manual / https://www.php.net/manual/en/index.php 5. Б.М.Голуб. С#. Концепція та синтаксис. Навчальний посібник. – Львів: ВНТЛ, 2016. – 144с. 6. CSS first steps / https://developer.mozilla.org/en-US/docs/Learn/CSS/First_steps
------------------------------------	--

	<ol style="list-style-type: none"> 7. Ruby – Вікіпідручник / https://uk.wikibooks.org/wiki/Ruby 8. Swift. Посібник з мови / https://book.swift.org.ua/book/1_language_guide 9. A Tour of Scala / https://docs.scala-lang.org/tutorials/ 10. Командний інтерпретатор Shell / http://ni.biz.ua/17/17_11/17_119813_komandniy-interpretator-SHELL.html 11. K.Seguin. The Little Go Book / https://www.openmymind.net/assets/go/go.pdf 12. Освоюємо R / https://uk.wikibooks.org/wiki/%D0%9E%D1%81%D0%B2%D0%BE%D1%8E%D1%94%D0%BC%D0%BE_R 13. TypeScript. Basic Types / https://www.typescriptlang.org/docs/handbook/basic-types.html 14. Kotlin docs / https://kotlinlang.org/docs/home.html 15. Introduction to Dart / https://dart.dev/language 16. TypeScript. Підручник / https://w3schoolsua.github.io/typescript/index.html#gsc.tab=0 17. Типи баз даних: особливості, відмінності та приклади / https://dou.ua/lenta/articles/types-of-databases/
Обсяг курсу	Загальний обсяг: 150 годин. Аудиторних занять: 48 год., з них 16 год. Лекцій та 32 години лабораторних робіт. Самостійної роботи: 102 год.
Очікувані результати навчання	<p>Після завершення цього курсу студент буде :</p> <p>Знати основи технологій: JavaScript, Java, Python, PHP, С#, CSS, Ruby, Swift, Scala, Shell, Go, Dart, TypeScript, Perl, Kotlin</p> <p>Вміти: застосовувати набуті знання для розробки програмного забезпечення.</p> <p>Курс забезпечує набуття таких компетентностей та програмних результатів навчання:</p> <p>Загальні компетентності (ЗК) і спеціальні (фахові, предметні) компетентності (СК):</p> <p>ЗК1. Здатність до абстрактного мислення, аналізу та синтезу.</p> <p>ЗК3. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.</p> <p>ЗК6. Здатність застосовувати теорію і методи Data Science для здійснення інтелектуального аналізу даних з метою виявлення нових властивостей та генерації нових знань про складні системи.</p>

	<p>ЗК7. Здатність проявляти лідерські навички, мотивувати людей, працювати в команді.</p> <p>ЗК8. Здатність свідомо та соціально-відповідально діяти на основі етичних міркувань і принципів академічної доброчесності.</p> <p>СК1. Здатність інтегрувати знання та здійснювати системні дослідження, застосовувати методи математичного та інформаційного моделювання складних систем та процесів різної природи.</p> <p>СК2. Здатність проєктувати архітектуру інформаційних систем.</p> <p>СК7. Здатність управляти робочими процесами у сфері інформаційних технологій, які є складними, непередбачуваними та потребують нових стратегічних підходів.</p> <p>СК8. Здатність розробляти і реалізовувати наукові та прикладні проєкти в галузі інформаційних технологій та дотичні до неї міждисциплінарні проєкти.</p>
--	---

Ключові слова	JavaScript, Java, Python, PHP, C#, CSS, Ruby, Swift, Scala, Shell, Go, Dart, TypeScript, Perl, Kotlin
Формат курсу	Очний
Теми	Подано нижче у таблиці Схеми курсу “Сучасні технології програмування”
Підсумковий контроль, форма	Екзамен
Пререквізити	Для вивчення курсу студенти потребують базових знань з - Програмування;
Навчальні методи та техніки, які будуть використовуватися під час викладання курсу	Презентації, лекції Індивідуальні завдання Групові проєкти, менторство
Необхідне обладнання	Комп’ютер та Internet-доступ до хмарного середовища (Amazon/Windows Azure/Google (App Engine)).
Критерії оцінювання (окремо для кожного виду навчальної діяльності)	<p>Оцінювання проводиться за 100-бальною шкалою.</p> <p>Впродовж семестру студент може отримати 50 балів. З них:</p> <p>- за роботу на лабораторних заняттях: максимальна кількість – 40 балів (4 програми (індивідуальні завдання) на вільний вибір (див. Теми курсу) по 10 балів); для кожного завдання встановлено терміни здачі. Роботи, які здаються із порушенням термінів без поважних причин, оцінюються на нижчу оцінку (кожне лабораторне заняття на 10% балів менше).</p> <p>- контрольна робота: максимальна кількість – 10 балів</p> <p>Підсумкова максимальна кількість балів 100.</p>

Поточне тестування та самостійна робота			Контр. робота	Екзамен	Сума
Індивідуальне завдання №1	...	Індивідуальне завдання №4	10	50	100
10	...	10			

Критерії оцінювання індивідуальних завдань:

Кожне лабораторне завдання має містити демонстрацію 10 функціональностей по вибраній темі (напр. оператори, функції, потоки, робота з файлами, серіалізація тощо)

10 балів – студент повністю виконав умови завдання, відповідає на всі запитання, пов'язані з тематикою завдання;

8-9 балів – студент повністю виконав умови завдання, на деякі запитання, пов'язані з тематикою завдання, відповідає з незначними неточностями;

6-7 балів – студент виконав завдання з незначними помилками, але самостійно їх виправляє, якщо на них вкаже викладач;

4-5 бали – студент виконав завдання частково (не менше 50%), проте демонструє вміння доповнення функціональності програми;

2-3 бали – студент виконав завдання частково (30-50%), проте демонструє вміння доповнення функціональності програми;

1 бал – студент виконав завдання частково (менше 30%), або не демонструє вміння доповнення функціональності програми;

0 балів – студент не виконав завдання.

Екзамен 50 балів. Екзамен проводиться у формі відповідей на питання білету (5 питань по 10 балів).

10 балів – студент повністю відповів на питання;

8-9 балів – студент відповів з незначними неточностями;

6-7 балів – студент відповів з помилками, але самостійно їх виправляє, якщо на них вкаже викладач;

4-5 бали – студент відповів частково (не менше 50%),

2-3 бали – студент відповів частково (30%-50%),

1 бал – студент відповів частково (менше 30%),

0 балів – студент не відповів на запитання.

Академічна доброчесність:

Відвідування лекцій та лабораторних завдань обов'язкове.

Жодні форми порушення академічної доброчесності не толеруються.

Питання до екзамену.	<ul style="list-style-type: none">- JavaScript- Java- Python- PHP- C#- CSS- Ruby- Swift- Scala- Shell- Go- Dart- TypeScript- PowerShell- Типи баз даних- Kotlin
Опитування	Анкету-оцінку з метою оцінювання якості курсу буде надано по завершенню курсу.

Схема курсу “Сучасні технології програмування”

Тиждень	Тема, план, короткі тези	Форма діяльності (заняття)	Література, ресурси в інтернеті	Завдання, год.	Термін виконання
1	Тема 1. Алгоритмічна мова JavaScript: <i>Структура мови. Ядро (ECMAScript). Об'єктна модель браузера (BOM). Об'єктна модель документа (DOM). Обробка подій. Відокремлення від розмітки. Букмарклетки. Користувацькі скрипти в браузері. Середовища розробки</i>	лекція (1 год)	[1]		
	Тема 2. Алгоритмічна мова Java: Синтаксис та ООП. Винятки. Графічний інтерфейс користувача. Аплети. Колекції. Менеджери розташування. Графічні компоненти Swing. Потоки. Паралелізм. Серіалізація. Середовища розробки	лекція (1 год)	[2]		
	Тема 1. Приклад JavaScript-програми	лабораторне (2 год)		Розробка JavaScript-програми (8 год.)	Під час заняття
2	Тема 2. Приклад Java-програми (Колекції)	лабораторне (2 год)		Розробка Java-програми (8 год.)	Під час заняття
3	Тема 3. Алгоритмічна мова Python: <i>Синтаксис та ООП. Інтерпретатор Пайтона. Обробка помилок. Виконання скриптів. Кодування джерела. Стартові файли для діалогового режиму. Середовища розробки</i>	лекція (1 год)	[3]		
	Тема 4. Алгоритмічна мова C#: Синтаксис C#. Простори імен. Модифікатори доступу. Структури і класи. Інтерфейси.	лекція (1 год)	[5]		

	Делегати. Події. Директиви препроцесору. Серіалізація. Середовища розробки				
	Тема 2. Приклад Java-програми (Потоки, Серіалізація)	лабораторне (2 год)		Розробка Java-програми (6 год.)	Під час заняття
4	Тема 3. Приклад Python-програми	лабораторне (2 год)		Розробка Python-програми (8 год.)	Під час заняття
5	Тема 5. Алгоритмічна мова TypeScript: Текстові редактори TS та Visual Studio Code. Статична типізація. Інтерфейси. Класи. Примітиви. Модулі. Файли-декларації. Середовища розробки	лекція (1 год)	[13,16]		
	Тема 6. Алгоритмічна мова PHP: Синтаксис та типи даних. Змінні, вирази та оператори. Масиви. Функції. HTML-форми та cookies. Класи та об'єкти. Робота з файлами. Робота з БД. Середовища розробки	лекція (1 год)	[4]		
	Тема 4. Приклад C#-програми	лабораторне (2 год)		Розробка C#-програми (6 год.)	Під час заняття
6	Тема 5. Приклад TypeScript-програми	лабораторне (2 год)		Розробка TypeScript-програми (8 год.)	Під час заняття
7	Тема 7. Алгоритмічна мова Kotlin: Функції, змінні, класи, трейти, зовнішні функції. Система типів. Функції вищих порядків. Предметно-орієнтовані мови. Середовища розробки.	лекція (1 год)	[14]		
	Тема 8. Бази даних. SQL: Типи баз даних. Ієрархічні БД. Мережеві БД. Реляційні (PostgreSQL, MS SQL, MySQL, MariaDB). Нереляційні БД. Документноорієнтовані	лекція (1 год)	[17]		

	(MongoDB). Графові (OrientDB)				
	Тема 6. Приклад PHP-програми	лабораторне (2 год)		Розробка PHP-програми (6 год.)	Під час заняття
8	Тема 7. Приклад Kotlin-програми	лабораторне (2 год)		Розробка Kotlin-програми (8 год.)	Під час заняття
9	Тема 9. Алгоритмічна мова Swift: Типи даних. Оператори. Кортежі. Функції. ООП у Swift. Колекції. Середовища розробки	лекція (1 год)	[8]		
	Тема 10. Алгоритмічна багатопотокова мова Go: Структура програми. Змінні, типи даних, оператори. Вказівники. Похідні типи. Пакети. Інтерфейси. Паралельне програмування. Мережеве програмування. Середовища розробки	лекція (1 год)	[11]		
	Тема 8. Приклади запитів у базах даних PostgreSQL та MongoDB	лабораторне (2 год)		Підготовка баз даних (6 год.)	Під час заняття
10	Тема 9. Приклад Swift-програми	лабораторне (2 год)		Розробка Swift-програми (8 год.)	Під час заняття
11	Тема 11. Алгоритмічна мова Ruby: Семантика. Контейнери. Процедурні об'єкти та ітератори. Класи, домішки, перевантаження операторів. Редактори Ruby. Стандартна бібліотека. Середовища розробки	лекція (1 год)	[7]		
	Тема 12. Алгоритмічна мова Dart: Крос-платформна мова Dart від Google. Основні конструкції, типи даних, алгоритмічні та функціональні можливості. Середовища розробки	лекція (1 год)	[15]		

	Тема 10. Приклад Go-програми	лабораторне (2 год)		Розробка Go-програми (6 год.)	Під час заняття
12	Тема 11. Приклад Ruby-програми	лабораторне (2 год)		Розробка Ruby-програми (8 год.)	Під час заняття
13	Тема 13. Алгоритмічна мова Scala: Ієрархія типів Scala. Класи, трейти, кортежі. Функції вищого порядку. Вкладені методи. Множинні списки параметрів. Регулярні вирази. Оператори. Анотації. Середовища розробки	лекція (1 год)	[9]		
	Тема 14. Мова опису зовнішнього вигляду сторінок CSS: Синтаксис CSS. Селектори та псевдокласи. Приєднання до сторінки. CSS рівня 2. Медіа-типи. Модель візуального форматування. Візуальні ефекти.	лекція (1 год)	[6]		
	Тема 12. Приклад Dart-програми	лабораторне (2 год)		Розробка Dart-програми (6 год.)	Під час заняття
14	Тема 13. Приклад Scala-програми	лабораторне (2 год)		Розробка Scala-програми (8 год.)	Під час заняття
15	Тема 15. Командний інтерпретатор Shell: Структура команд. Групування команд. Генерація імен файлів. Shell-змінні. Параметри. Програмне середовище. Програмні структури. Функції в Shell.	лекція (2 год)	[10]		
	Тема 14. Приклади функціональності CSS	лабораторне (2 год)			Під час заняття
16	Тема 15. Приклади функціональності Shell	лабораторне (2 год)		Підготовка демонстраційного коду Shell (2 год.)	Під час заняття