

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

Оновлено та затверджено
на засіданні
кафедри обчислювальної математики
факультету прикладної математики та
інформатики
Львівського національного університету
імені Івана Франка
(протокол № 6 від 21 грудня 2023 р.)



Завідувач кафедри

Роман ХАПКО

Силабус з навчальної дисципліни
«Сучасні технології програмування (англійською мовою)»,
що викладається в межах ОНП Прикладна математика
другого (магістерського) рівня вищої освіти для здобувачів з
спеціальності 113 – прикладна математика

Львів 2023 р.

Назва дисципліни	Сучасні технології програмування (англійською мовою)
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 обчислювальної математики
Галузь знань, шифр та назва спеціальності	11 – математика і статистика 113 – прикладна математика
Викладачі дисципліни	Вавричук Василь Григорович, кандидат фізико-математичних наук, доцент кафедри обчислювальної математики
Контактна інформація викладачів	vasyi.vavrychuk@lnu.edu.ua, https://ami.lnu.edu.ua/employee/vavrychuk Головний корпус ЛНУ ім. І. Франка, каб. 360. м. Львів, вул. Університетська, 1
Консультації з питань навчання по дисципліні відбуваються	Консультації в день проведення лекцій/лабораторних занять, за розкладом консультацій кафедри, а також в середовищі Microsoft Teams.
Сторінка курсу	https://ami.lnu.edu.ua/course/suchasni-tekhnologii-prohramuvannia-anhliyskoii-movoii-trykkladna-matematyka-1-9
Інформація про дисципліну	Дисципліна «Сучасні технології програмування» є нормативною дисципліною для спеціальності 113 - прикладна математика для ОНП «Прикладна математика», яка викладається в 3-му семестрі в обсязі 4,5 кредитів (за Європейською Кредитно-Трансферною Системою ECTS).
Коротка анотація дисципліни	ІТ індустрія на даний час є однією з найважливіших галузей виробництва в Україні. Не зважаючи на те, що в ІТ сфері функціонує певне коло професійних занять, ядром ІТ сфери можна вважати програмування. У цьому курсі приділена увага основним напрямкам в програмуванні та сучасним тенденціям вирішення задач, які постають в цих напрямках.
Мета та цілі дисципліни	Метою дисципліни є розвиток у студентів знань та вмінь з програмування. Ознайомлення з основними напрямками та сучасними тенденціями в програмуванні.
Література для вивчення дисципліни	Основна література: <ol style="list-style-type: none"> 1. The Practice of Programming, Brian W. Kernighan, Rob Pike, Addison-Wesley Professional, February 1999 2. The Pragmatic Programmer: your journey to mastery, 20th Anniversary Edition, 2nd Edition, David Thomas, Andrew Hunt, Addison-Wesley Professional, September 2019 3. Microservices Patterns, Chris Richardson, Manning Publications, November 2018 4. Building Microservices, 2nd Edition, Sam Newman, O'Reilly Media, Inc., August 2021 5. A Tour of C++, 3rd Edition, Bjarne Stroustrup, Addison-Wesley Professional, September 2022 Додаткова література: <ol style="list-style-type: none"> 6. https://www.oracle.com/java/technologies/javase/javase-tech-

	database.html 7. https://jdbi.org 8. https://www.npgsql.org 9. https://spring.io 10. https://developer.mozilla.org/en-US/ 11. https://ionicframework.com 12. https://flutter.dev 13. https://systemd.io 14. https://www.coursera.org/learn/intro-fpga-design-embedded-systems 15. https://www.coursera.org/learn/fpga-hardware-description-languages 16. https://people.ece.cornell.edu/land/courses/ece5760/ 17. Harris & Harris. Digital Design and Computer Architecture. 18. From Nand to Tetris, Building a Modern Computer From First Principles, https://www.nand2tetris.org
Обсяг курсу	Загальний обсяг: 135 годин. Аудиторних занять: 48 год., з них 16 год. лекцій та 32 год. лабораторних робіт. Самостійної роботи: 87 год.
Очікувані результати навчання	Після завершення цього курсу студент буде знати: <ul style="list-style-type: none"> • основні спеціалізації в програмуванні та задачі, які перед ними стоять • актуальні мови програмування та фреймворки в різних галузях програмування вміти: <ul style="list-style-type: none"> • вирішувати стандартні задачі в програмуванні, використовуючи актуальні підходи • користуватися інструментами поширеними у різних колах • відлагоджувати баги, які виникають в процесі роботи Курс забезпечує набуття таких компетентностей: Загальні компетентності: <ul style="list-style-type: none"> • ЗК02. Здатність застосовувати знання в практичних ситуаціях. • ЗК04. Здатність генерувати нові ідеї (креативність). • ЗК06. Здатність спілкуватися іноземною мовою з професійних питань як усно, так і письмово. Фахові компетентності спеціальності: <ul style="list-style-type: none"> • ФК04. Знання сучасних технологій обчислень та програмування. • ФК05. Здатність розробляти та оптимізувати комп'ютерні програми зі складною логікою. • ФК06. Здатність розробляти проекти та управляти ними. та програмних результатів навчання: <ul style="list-style-type: none"> • ПРН03. Розробляти та програмно реалізувати алгоритми розв'язування складних прикладних задач. • ПРН05. Поєднувати алгоритми та структури даних для вирішення практичних задач. • ПРН06. Спілкуватися іноземною мовою як усно, так і письмово при обговоренні професійних питань та проведенні досліджень.
Ключові слова	ІТ, програмування, прикладна математика
Формат курсу	Очний
Теми	Подано нижче у таблиці «Схема курсу»
Підсумковий контроль, форма	Екзамен.
Пререквізити	Для вивчення курсу студенти потребують базових знань з програмування,

	управління проектами та знань з іноземної мова за професійним спрямуванням.																															
Навчальні методи та техніки, які будуть використовуватися під час викладання курсу	Презентації, лекції, модульний контроль, індивідуальні завдання.																															
Необхідне обладнання	Комп'ютер із програмним забезпеченням Visual Studio, Eclipse, Node.js, .NET, WSL, Docker, Git, Internet.																															
Критерії оцінювання (окремо для кожного виду навчальної діяльності)	<p>Оцінювання проводиться за 100-бальною шкалою.</p> <table border="1"> <thead> <tr> <th colspan="2">Оцінка за шкалою ECTS</th> <th>Оцінка в балах</th> <th colspan="2">Оцінка за національною шкалою</th> </tr> </thead> <tbody> <tr> <td>A</td> <td>Відмінно</td> <td>100 - 90</td> <td>Відмінно</td> <td>5</td> </tr> <tr> <td>B</td> <td>Дуже добре</td> <td>81- 89</td> <td rowspan="2">Добре</td> <td rowspan="2">4</td> </tr> <tr> <td>C</td> <td>Добре</td> <td>71 -80</td> </tr> <tr> <td>D</td> <td>Задовільно</td> <td>61 - 70</td> <td rowspan="2">Задовільно</td> <td rowspan="2">3</td> </tr> <tr> <td>E</td> <td>Достатньо</td> <td>51- 60</td> </tr> <tr> <td>FX (F)</td> <td>Незадовільно</td> <td>0 - 50</td> <td>Незадовільно</td> <td>2</td> </tr> </tbody> </table> <p>Під час семестру студент може отримати 50 балів з них:</p> <ul style="list-style-type: none"> • 10 балів за індивідуальні завдання №1-4 та доповідь. <p>На екзамені студент може отримати 50 балів.</p> <p>Критерії оцінювання індивідуальних завдань:</p> <p>10 балів - Програма повністю реалізовує необхідний функціонал, осмислено спроектована з використанням архітектурних патернів, патернів проектування, принципів парадигми програмування, ідіом; імена в програмі є осмисленими, підтримується єдиний стиль кодування. Студент компетентно пояснює програму та може внести зміни.</p> <p>8-9 балів - Програма не повно реалізовує необхідний функціонал. У архітектуру та код можна внести покращення. Студент компетентно пояснює програму та може внести зміни.</p> <p>5-7 балів - Програма реалізовує більшість необхідного функціоналу. Архітектура та код можуть бути дещо хаотичними. Студент може пояснити програму, але з внесенням змін можуть бути труднощі.</p> <p>2-4 бали - Програма реалізовує найпростіший юзкейс. Студент може пояснити програму, але з внесенням змін можуть бути труднощі.</p> <p>1 бал - Програма наявна, студент може її запустити та продемонструвати, проте пояснення її роботи є незадовільним.</p> <p>0 балів - програми немає або вона не запускається.</p> <p>Критерії оцінювання доповіді та відповідей на екзамені:</p> <p>10 балів - Тема висвітлюється повністю та цілісно, логічно та структуровано, враховуючи пов'язаний з нею контекст, причинно-наслідковід зв'язки. Наводяться приклади, покрокові інструкції. Впевнено відповідає на всі запитання, які пов'язані з тематикою доповіді та демонструє глибокі знання.</p> <p>8-9 балів - Тема висвітлюється майже повністю та з незначними</p>	Оцінка за шкалою ECTS		Оцінка в балах	Оцінка за національною шкалою		A	Відмінно	100 - 90	Відмінно	5	B	Дуже добре	81- 89	Добре	4	C	Добре	71 -80	D	Задовільно	61 - 70	Задовільно	3	E	Достатньо	51- 60	FX (F)	Незадовільно	0 - 50	Незадовільно	2
Оцінка за шкалою ECTS		Оцінка в балах	Оцінка за національною шкалою																													
A	Відмінно	100 - 90	Відмінно	5																												
B	Дуже добре	81- 89	Добре	4																												
C	Добре	71 -80																														
D	Задовільно	61 - 70	Задовільно	3																												
E	Достатньо	51- 60																														
FX (F)	Незадовільно	0 - 50	Незадовільно	2																												

	<p>неточностями. Відповідає на більшість запитань, які пов'язані з тематикою доповіді.</p> <p>4-7 балів - Тема висвітлюється не повністю та помилками, проте їх меншість та студент торкнувся основних понять.</p> <p>1-3 бали - Студент володіє лише найпростішими поняттями з даної теми.</p> <p>0 балів - Студент не підготував доповіді або тему доповіді не розкрито і під час захисту студент не може відповісти на жодне запитання за тематикою доповіді.</p> <p>Академічна доброчесність: Очікується, що роботи студентів будуть їх оригінальними дослідженнями чи міркуваннями. Відсутність посилань на використані джерела, фабрикування джерел, списування, втручання в роботу інших студентів становлять, але не обмежують, приклади можливої академічної недоброчесності. Виявлення ознак академічної недоброчесності в письмовій роботі студента є підставою для її незарахування викладачем, незалежно від масштабів плагіату чи обману.</p> <p>Відвідання занять є важливою складовою навчання. Очікується, що всі студенти відвідають усі лекції та лабораторні заняття курсу. Студенти повинні інформувати викладача про неможливість відвідати заняття. У будь-якому випадку студенти зобов'язані дотримуватися термінів визначених для виконання всіх видів письмових робіт та індивідуальних завдань, передбачених курсом.</p> <p>Література. Уся література, яку студенти не зможуть знайти самостійно, буде надана викладачем виключно в освітніх цілях без права її передачі третім особам. Студенти заохочуються до використання також й іншої літератури та джерел, яких немає серед рекомендованих.</p> <p>Політика виставлення балів. Враховуються бали набрані при здачі та захисті індивідуальних завдань та бали за іспит. При цьому обов'язково враховуються присутність на заняттях та активність студента під час лабораторного заняття; недопустимість пропусків та запізнь на заняття; користування мобільним телефоном, планшетом чи іншими мобільними пристроями під час заняття в цілях не пов'язаних з навчанням; списування та плагіат; несвоєчасне виконання поставленого завдання і т. ін. Жодні форми порушення академічної доброчесності не толеруються.</p>
<p>Питання до екзамену.</p>	<ol style="list-style-type: none"> 1. Нативні клієнти баз даних. 2. Засоби ORM. 3. Реалізація REST сервісу. 4. Засоби документування та тестування API: Swagger та ін. 5. Специфікація OpenAPI. 6. Засоби інверсії контролю та ін'єкції залежностей. 7. Способи автентифікації. 8. DOM інтерфейс. 9. AJAX технологія. 10. Фреймворки крос-платформних мобільних додатків: Ionic та ін. 11. Фреймворк створення додатків Flutter. 12. Програмування в режимі ядра. 13. Визначення системних systemd сервісів. 14. Запуск Yocto на емуляторі QEMU. 15. Схемотехнічний підхід розробки апаратного забезпечення. 16. Комбінаційна логіка. 17. Послідовна логіка. 18. Мови опису апаратного забезпечення.
<p>Опитування</p>	<p>Анкету-оцінку з метою оцінювання якості курсу буде надано по завер-</p>

Схема курсу «Сучасні технології програмування»

Ти ж.	Тема, план, короткі тези	Форма діяльності (заняття)	Література. Ресурси інтернеті	Завдання, год.	Термін виконання
1	Тема 1. Вступ Спеціалізації в програмуванні.	Лекція (2 год.)	[1-2]	Опрацювання лекційного матеріалу (3 год.)	1 тиждень
	Тема 1. Вступ Налаштування робочого середовища, WSL. Індивідуальне завдання №1: Інтерфейс CRUD операцій з даними.	Лабораторна робота (2 год.)	https://learn.microsoft.com/en-us/windows/wsl/	Виконання індивідуального завдання №1 (3 год.)	1 тиждень
2	Тема 1. Вступ Обговорення стеків технологій, дорожніх карт, тем доповідей.	Лабораторна робота (2 год.)	https://roadmap.sh - Developer Roadmaps https://awesome.re - Awesome lists about all kinds of interesting topics	Виконання індивідуального завдання №1 (3 год.)	1 тиждень
3	Тема 2. Технології розробки бекенд Нативні клієнти БД, засоби ORM, REST, інверсія контролю, ін'єкції залежностей.	Лекція (2 год.)	[3-4], [6-9]	Опрацювання лекційного матеріалу (3 год.)	1 тиждень
	Тема 2. Технології розробки бекенд Демо проекти.	Лабораторна робота (2 год.)	[3-4], [6-9]	Виконання індивідуального завдання №1 (5 год.)	1 тиждень
4	Тема 2. Технології розробки бекенд Доповіді студентів.	Лабораторна робота (2 год.)	[3-4], [6-9]	Виконання індивідуального завдання №1 (5 год.)	1 тиждень
5	Тема 3. Технології розробки фронтенд DOM інтерфейс, AJAX технологія.	Лекція (2 год.)	[10]	Опрацювання лекційного матеріалу (3 год.)	1 тиждень

	Тема 3. Технології розробки фронтенд Тест, демо проекти.	Лабораторна робота (2 год.)	[10]	Виконання індивідуального завдання №1 (3 год.)	1 тиждень
6	Тема 3. Технології розробки фронтенд Доповіді студентів.	Лабораторна робота (2 год.)	[10]	Виконання індивідуального завдання №1 (3 год.)	1 тиждень
7	Тема 4. Мобільна розробка Ionic, Flutter.	Лекція (2 год.)	[11-12]	Опрацювання лекційного матеріалу (3 год.)	1 тиждень
	Тема 4. Мобільна розробка Тест, демо проекти.	Лабораторна робота (2 год.)	[11-12]	Виконання індивідуального завдання №1 (3 год.)	1 тиждень
8	Тема 4. Мобільна розробка Доповіді студентів.	Лабораторна робота (2 год.)	[11-12]	Виконання індивідуального завдання №1 (3 год.)	1 тиждень
9	Тема 5. Системне програмування Systemd, IPC, gRPC, WebSockets.	Лекція (2 год.)	[5], [13]	Опрацювання лекційного матеріалу (3 год.)	1 тиждень
	Тема 5. Системне програмування Здача індивідуального завдання №1. Індивідуальне завдання №2: Системний сервіс.	Лабораторна робота (2 год.)	[5], [13]	Виконання індивідуального завдання №2 (5 год.)	1 тиждень
10	Тема 5. Системне програмування Доповіді студентів.	Лабораторна робота (2 год.)	[5], [13]	Виконання індивідуального завдання №2 (5 год.)	1 тиждень
11	Тема 6. Вбудовані системи Yocto, QEMU, архітектура ARM, ядро.	Лекція (2 год.)	https://www.yoctoproject.org	Опрацювання лекційного матеріалу (3 год.)	1 тиждень
	Тема 6. Вбудовані системи Здача індивідуального	Лабораторна робота (2 год.)	https://www.yoctoproject.org	Виконання індивідуального завдання №3 (5 год.)	1 тиждень

	завдання №2. Індивідуальне завдання №3: Інтеграція у вбудовану систему.				
12	Тема 6. Вбудовані системи Доповіді студентів.	Лабораторна робота (2 год.)	https://www.yoctoproject.org	Виконання індивідуального завдання №3 (5 год.)	1 тиждень
13	Тема 7. Розробка апаратного забезпечення Intel Quartus, Icarus Verilog, Verilator, GtkWave.	Лекція (2 год.)	[14-18]	Опрацювання лекційного матеріалу (3 год.)	1 тиждень
	Тема 7. Розробка апаратного забезпечення Здача індивідуального завдання №3. Індивідуальне завдання №4: Апаратна реалізація компоненти цифрової логіки.	Лабораторна робота (2 год.)	[14-18]	Виконання індивідуального завдання №4 (5 год.)	1 тиждень
14	Тема 7. Розробка апаратного забезпечення Доповіді студентів.	Лабораторна робота (2 год.)	[14-18]	Виконання індивідуального завдання №4 (5 год.)	1 тиждень
15	Підсумкова лекція Огляд альтернативних технологій, бібліотек, мов програмування і т.д.	Лекція (2 год.)	https://roadmap.sh - Developer Roadmaps https://awesome.re - Awesome lists about all kinds of interesting topics	Опрацювання лекційного матеріалу, виконання індивідуального завдання (5 год.)	1 тиждень
	Тема 7. Розробка апаратного забезпечення Проектування комбінаційної логіки,	Лабораторна робота (2 год.)	[14-18]	Опрацювання лекційного матеріалу, виконання індивідуального завдання	1 тиждень

	послідовної логіки, цифрових функціональних вузлів.			(3 год.)	
16	Тема 7. Розробка апаратного забезпечення Здача індивідуального завдання №4.	Лабораторна робота (2 год.)	[14-18]	Повторення, погашення технічного боргу	

Discipline name	Modern programming technologies
Address where the discipline is taught	The main building of Ivan Franko National University of Lviv 1, Universytetska st., Lviv
The faculty and department under which the discipline is established	Faculty of Applied Mathematics and Informatics Department of Computational Mathematics
Field of knowledge, code and name of specialty	11 – mathematics and statistics 113 – applied mathematics
Teachers of the discipline	Vavrychuk Vasyl Hryhorovych, Candidate of Physical and Mathematical Sciences, Associate Professor of the Department of Computational Mathematics,
Contact information of teachers	vasyi.vavrychuk@lnu.edu.ua, https://ami.lnu.edu.ua/employee/vavrychuk Room 262, The main building of Ivan Franko National University of Lviv, 1, Universytetska st., Lviv
Consultations on issues of training in the discipline are taking place	Consultations on the day of lectures/laboratory sessions, by the schedule of the chair, and in the Microsoft Teams environment.
Course page	https://ami.lnu.edu.ua/course/suchasni-tekhnohii-prohramuvannia-anhliyskoiu-movoju-prykladna-matematyka
Information about the discipline	The discipline "Models of deep machine learning" is a mandatory discipline for the specialty 113 - applied mathematics for the educational program "Applied Mathematics", which is taught in the 3st semester in the amount of 4.5 credits (according to the European Credit Transfer System ECTS).
Brief abstract of the discipline	The IT industry is currently one of the most important branches of the economy in Ukraine. Despite the fact that a certain range of professional occupations functions in the IT sphere, programming can be considered the core of the IT field. In this course, attention is paid to the main directions in programming and modern trends in solving problems that appear in these directions.
Goal and objectives of the discipline	The goal of the discipline is the development of students' knowledge and skills in programming. Acquaintance with the main directions and modern trends in programming.
Literature for the discipline	Basic literature: <ol style="list-style-type: none"> 1. The Practice of Programming, Brian W. Kernighan, Rob Pike, Addison-Wesley Professional, February 1999 2. The Pragmatic Programmer: your journey to mastery, 20th Anniversary Edition, 2nd Edition, David Thomas, Andrew Hunt, Addison-Wesley Professional, September 2019 3. Microservices Patterns, Chris Richardson, Manning Publications, November 2018 4. Building Microservices, 2nd Edition, Sam Newman, O'Reilly Media, Inc., August 2021 5. A Tour of C++, 3rd Edition, Bjarne Stroustrup, Addison-Wesley Professional, September 2022 Additional literature: <ol style="list-style-type: none"> 6. https://www.oracle.com/java/technologies/javase/javase-tech-

	database.html 7. https://jdbi.org 8. https://www.npgsql.org 9. https://spring.io 10. https://developer.mozilla.org/en-US/ 11. https://ionicframework.com 12. https://flutter.dev 13. https://systemd.io 14. https://www.coursera.org/learn/intro-fpga-design-embedded-systems 15. https://www.coursera.org/learn/fpga-hardware-description-languages 16. https://people.ece.cornell.edu/land/courses/ece5760/
Scope of the course	Total volume: 135 hours. Classroom classes: 48 hours, including 16 hours of lectures and 32 hours of laboratory work. Independent work: 87 hours.
Expected learning outcomes	<p>After completing this course student will know</p> <ul style="list-style-type: none"> the main specializations in programming and the challenges they face up to date programming languages and frameworks in various fields of programming <p>be able to:</p> <ul style="list-style-type: none"> solve standard problems in programming using well-known approaches use tools common in various areas debug bugs that arise during work <p>The course ensures the acquisition of the following competencies:</p> <p>General competences:</p> <ul style="list-style-type: none"> GC 02. Ability to apply theoretical knowledge in practical cases. GC 04. Ability to generate new ideas (creativity). GC 06. Ability to communicate in a foreign language on professional matters both orally and in writing. <p>Professional competencies of the specialty:</p> <ul style="list-style-type: none"> PK04. Knowledge of modern computing and programming technologies. PK05. Ability to debug and optimize computer programs with complex logic. PK06. Ability to develop and manage projects. <p>and program learning outcomes:</p> <ul style="list-style-type: none"> PLO 03. Develop and programmatically implement algorithms for solving complex applied problems. PLO 05. Combine algorithms and data structures to solve practical problems. PLO 06. Communicate freely in a foreign language both orally and in writing when discussing professional issues and conducting research.
Keywords	IT, programming, applied mathematics
Course format	Face-to-face
Topics	See below in the table Scheme of the course
Final control,	Exam at the end of the semester.

form																											
Prerequisites	To study the course, students need basic knowledge of programming.																										
Teaching methods -and techniques that will be used during the teaching of the course	Presentations, lectures, modular control. Homework and individual tasks.																										
Required equipment	Computer, Internet.																										
Evaluation criteria - (separately for each type of educational activity)	The assessment is carried out on a 100-point scale.																										
	<table border="1"> <thead> <tr> <th colspan="2">Evaluation on the ECTS scale</th> <th>Score in points</th> <th>Evaluation on a national scale</th> </tr> </thead> <tbody> <tr> <td>A</td> <td>Excelent</td> <td>100-90</td> <td>Excelent</td> </tr> <tr> <td>B</td> <td>Very good</td> <td>81-89</td> <td rowspan="2">Good</td> </tr> <tr> <td>C</td> <td>Good</td> <td>71-80</td> </tr> <tr> <td>D</td> <td>Satisfactory</td> <td>61-70</td> <td rowspan="2">Satisfactory</td> </tr> <tr> <td>E</td> <td>Enough</td> <td>51-60</td> </tr> <tr> <td>FX (F)</td> <td>Non-satisfactory</td> <td>0-50</td> <td>Non-satisfactory</td> </tr> </tbody> </table>	Evaluation on the ECTS scale		Score in points	Evaluation on a national scale	A	Excelent	100-90	Excelent	B	Very good	81-89	Good	C	Good	71-80	D	Satisfactory	61-70	Satisfactory	E	Enough	51-60	FX (F)	Non-satisfactory	0-50	Non-satisfactory
	Evaluation on the ECTS scale		Score in points	Evaluation on a national scale																							
	A	Excelent	100-90	Excelent																							
	B	Very good	81-89	Good																							
	C	Good	71-80																								
	D	Satisfactory	61-70	Satisfactory																							
E	Enough	51-60																									
FX (F)	Non-satisfactory	0-50	Non-satisfactory																								
During the semester, a student can receive 50 points from them:																											
<ul style="list-style-type: none"> • 10 points for individual tasks #1-4 and report. 																											
A student can get 50 points on the exam.																											
<p>Evaluation criteria for individual tasks:</p> <p>10 points - The program fully implements the necessary functionality, is meaningfully designed using architectural patterns, design patterns, programming paradigm principles, idioms; names in the program are meaningful, a single coding style is supported. The student competently explains the program and can make changes.</p> <p>8-9 points - The program does not fully implement the necessary functionality. Improvements can be made to the architecture and code. The student competently explains the program and can make changes.</p> <p>5-7 points - The program implements most of the necessary functionality. The architecture and code can be somewhat chaotic. The student can explain the program, but there may be difficulties in making changes.</p> <p>2-4 points - The program implements the simplest use case. The student can explain the program, but there may be difficulties in making changes.</p> <p>1 point - The program exists, the student can run it and demonstrate it, but the explanation of its operation is unsatisfactory.</p> <p>0 points - there is no program or it does not start.</p> <p>Criteria for evaluating the report and exam answers:</p> <p>10 points - The topic is covered completely and holistically, logically and structured, taking into account the related context, cause-and-effect relationships. Examples and step-by-step instructions are provided. Confidently</p>																											

	<p>answers all questions related to the topic of the report and demonstrates deep knowledge.</p> <p>8-9 points - The topic is covered almost completely and with minor inaccuracies. Answers most questions related to the topic of the report.</p> <p>4-7 points - The topic is covered incompletely and with errors, but their minority and the student touched on the main concepts.</p> <p>1-3 points - the student has only the simplest concepts on this topic.</p> <p>0 points - the student has not prepared a report or the topic of the report has not been disclosed and during the defense the student cannot answer any question on the topic of the report.</p> <p>Academic Integrity: Students' works are expected to be their own original research or reasoning. Lack of references to used sources, fabrication of sources, plagiarism, interference in the work of other students are, but are not limited to, examples of possible -academic dishonesty. The detection of signs of academic dishonesty in a student-'s written work is a reason for its rejection by -the teacher, regardless of the scale of plagiarism or deception.</p> <p>Attending classes is an important part of learning. All students are expected to attend all lectures and laboratory sessions of the course. Students must inform the teacher about the impossibility to attend classes. In any case, students are obliged to adhere to the deadlines -set for homework and individual assignments provided by the course.</p> <p>Literature. All literature that students cannot find on their own will be provided by the teacher for educational purposes only, without the right to transfer it to third parties. Students are also encouraged to use other literature and sources that are not among the recommended ones.</p> <p>Scoring policy. Points scored on laboratory classes (homeworks and individual tasks) and during testing are taken into account. At the same time, attendance of classes and the student's activity during practical sessions must be taken into account; inadmissibility of absences and lateness to classes; using a mobile phone, tablet or other mobile -devices during class for purposes not related to education; plagiarism; untimely performance of the assigned task, etc.</p> <p>Any form of breach of academic integrity will not be tolerated.</p>
<p>Questions for exam</p>	<ol style="list-style-type: none"> 1. Native database clients. 2. ORM tools. 3. Implementation of REST service. 4. API documentation and testing tools: Swagger et al. 5. OpenAPI specification. 6. Inversion of control and dependency injection 7. Authentication methods. 8. DOM interface. 9. AJAX technology. 10. Cross-platform mobile application frameworks: Ionic and others. 11. Flutter application creation framework. 12. Kernel mode programming. 13. Definition of system systemd services. 14. Running Yocto on the QEMU emulator.

	15. Schematic approach to hardware development. 16. Implementation of combinatorial logic. 17. Implementation of sequential logic. 18. Hardware description languages.
Poll	An evaluation questionnaire for the purpose of assessing the quality of the course will be provided upon completion -of the course.

Scheme of the course “Modern programming technologies”

Week	Topic, plan, short theses	Form of activity (class)	Literature, Internet resources	Assignment, hours	Deadline
1	Topic 1. Introduction Specialization in programming.	Lecture (2 hours)	[1-2]	Processing of lecture material (3 hours)	1 week
	Topic 1. Introduction Configure work environment, WSL. Individual task №1: Interface for CRUD operations with data.	Laboratory (2 hours)	https://learn.microsoft.com/en-us/windows/wsl/	Doing task №1 (3 hours)	1 week
2	Topic 1. Introduction Discussion of technology stacks, roadmaps, report topics.	Laboratory (2 hours)	https://roadmap.sh - Developer Roadmaps https://awesome.re - Awesome lists about all kinds of interesting topics	Processing of lecture material (3 hours)	1 week
3	Topic 2. Backend development Native DB clients, ORM tools, REST, inversion of control, dependency injection.	Lecture (2 hours)	[3-4], [6-9]	Processing of lecture material (3 hours)	1 week
	Topic 2. Backend development Implementation of a demo	Laboratory (2 hours)	[3-4], [6-9]	Doing task №1 (5 hours)	1 week

	project.				
4	Topic 2. Backend development Presentations of students.	Laboratory (2 hours)		Doing task №1 (5 hours)	1 week
5	Topic 3. Frontend development DOM interface, AJAX technology.	Lecture (2 hours)	[10]	Processing of lecture material (3 hours)	1 week
	Topic 3. Frontend development Implementation of a demo project.	Laboratory (2 hours)	[10]	Doing task №1 (3 hours)	1 week
6	Topic 3. Frontend development Presentations of students.	Laboratory (2 hours)		Doing task №1 (3 hours)	1 week
7	Topic 4. Mobile development Ionic, Flutter.	Lecture (2 hours)	[11-12]	Processing of lecture material (3 hours)	1 week
	Topic 4. Mobile development Implementation of a demo project.	Laboratory (2 hours)	[11-12]	Doing task №1 (3 hours)	1 week
8	Topic 4. Mobile development Presentations of students.			Doing task №1 (3 hours)	1 week
9	Topic 5. System programming Systemd, IPC, gRPC, WebSockets.	Lecture (2 hours)	[5], [13]	Processing of lecture material (3 hours)	1 week
	Topic 5. System	Laboratory	[5], [13]	Doing task №2 (5 hours)	1 week

	programming Submission of individual task №1. Individual task №2: System service.	(2 hours)		hours)	
10	Topic 5. System programming Presentations of students.	Laboratory (2 hours)		Doing task №2 (5 hours)	1 week
11	Topic 6. Emebedded systems Yocto, QEMU, architecture ARM, kernel.	Lecture (2 hours)	https://www.yoctoproject.org	Processing of lecture material (3 hours)	1 week
	Topic 6. Emebedded systems Submission of individual task №2. Individual task №3: Integration in embedded system.	Laboratory (2 hours)	https://www.yoctoproject.org	Doing task №3 (5 hours)	1 week
12	Topic 6. Emebedded systems Presentations of students.	Laboratory (2 hours)		Doing task №3 (5 hours)	1 week
13	Topic 7. Hardware development Intel Quartus, Icarus Verilog, GtkWave.	Lecture (2 hours)	[14-16]	Processing of lecture material (3 hours)	1 week
	Topic 7. Hardware development Submission of individual task	Laboratory (2 hours)	[14-16]	Doing task №4 (5 hours)	1 week

	№3. Individual task №4: Hardware implementation of digital logic.				
14	Topic 7. Hardware development Presentations of students.	Laboratory (2 hours)		Doing task №4 (5 hours)	1 week
15	Final lecture Overview of alternative technologies, libraries, programming languages, etc.	Lecture (2 hours)		Processing of lecture material and doing task №4 (5 hours)	1 week
	Submission of individual task №4.	Laboratory (2 hours)		Processing of lecture material and doing task №4 (3 hours)	1 week
16	Topic 7. Hardware development Submission of individual task №4.	Laboratory (2 hours)		Repetition, reducing technical dept.	