

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Львівський національний університет імені Івана Франка
Факультет електроніки та комп'ютерних технологій
Кафедра радіоелектронних і комп'ютерних систем

Затверджено

На засіданні
кафедри радіоелектронних і
комп'ютерних систем
факультету електроніки та комп'ютерних
технологій
Львівського національного університету
імені Івана Франка
(протокол № 13 від 31 серпня 2022 р.)

Завідувач кафедри:



Ігор ОЛЕНИЧ

Силабус з навчальної дисципліни
“Системне програмування”,
що викладається в межах ОПП
“ Високопродуктивний комп'ютинг ”
першого (бакалаврського) рівня вищої освіти для здобувачів з
спеціальності 121 – Інженерія програмного забезпечення

Львів 2022 р.

Назва дисципліни	Системне програмування
Адреса викладання дисципліни	Корпус факультету електроніки та комп'ютерних технологій, Львівський національний університет імені Івана Франка, вул. Драгоманова 50, м. Львів, 79005, вул. Ген. Тарнавського 107, м. Львів, 79011
Факультет та кафедра, за якою закріплена дисципліна	Факультет електроніки та комп'ютерних технологій, кафедра радіоелектронних і комп'ютерних систем
Галузь знань, шифр та назва спеціальності	12 Інформаційні технології, 121 Інженерія програмного забезпечення
Викладачі дисципліни	Бойко Ярослав Васильович, канд. фіз.-мат. наук, доцент
Контактна інформація викладачів	yaroslav.bojko@lnu.edu.ua, https://electronics.lnu.edu.ua/employee/bojko-ya-v
Консультації з питань навчання по дисципліні відбуваються	Консультації в день проведення лекційних занять (за попередньою домовленістю). Також можливі он-лайн консультації через MS Teams. Для погодження часу он-лайн консультацій слід писати на електронну пошту викладача.
Сторінка дисципліни	https://e-learning.lnu.edu.ua/course/view.php?id=4296
Інформація про дисципліну	Дисципліна «Системне програмування» є дисципліною циклу професійної та практичної підготовки зі спеціальності 121 Інженерія програмного забезпечення для освітньої програми «Високопродуктивний комп'ютинг», яка викладається в 3 семестрі в обсязі 6,0 кредитів (за Європейською Кредитно-Трансферною Системою ECTS).
Коротка анотація дисципліни	Предметом вивчення навчальної дисципліни є технології програмного керування комп'ютерними системами. Програма навчальної дисципліни складається із трьох модулів і трьох змістових модулів — Високорівневі засоби системного програмування, Низькорівневе системне програмування, Сучасні мови системного програмування. Особливістю дисципліни є детальне вивчення актуальних технологій і програмних засобів системного програмування у різних галузях інформаційних технологій і програмної інженерії.
Мета та цілі дисципліни	Метою вивчення дисципліни є формування у студентів знань про останні досягнення в галузі технологій створення системного програмного забезпечення і формування у студентів практичних навичок професійного використання високо- та низькорівневих засобів програмування систем, включно зі створенням програмного коду режиму ядра ОС.
Література для вивчення дисципліни	<p>Основна література:</p> <ol style="list-style-type: none"> 1. Pya Shpigor. Bash programming from scratch. – Leanpub, – 2021. – 320 p. 2. Mark G. Sobell, Matthew Helmke. A Practical Guide to Linux Commands, Editors, and Shell Programming, 4th Edition. – Addison-Wesley Professional. – 2017. – 1232 p. 3. Chris Dent, Mastering Windows PowerShell Scripting. Third Edition. – Packt. – 2019. – 606 p. 4. Thomas Lee. PowerShell 7 for IT Pros. A Guide to Using PowerShell 7 to Manage Windows® Systems. – Wiley. – 2021. – 465 p. 5. Richard Bullington-McGuire, Andrew K. Dennis, Michael Schwartz. Docker for Developers. – Packt. – 2020. – 468 p. 6. Ian Miell, Aidan Hobson Sayers. Docker in Practice. Second Edition. – Manning Publications. – 2019. – 456 p. 7. Rodolfo Giometti. Linux Device Driver Development Cookbook. – Packt. – 2019. – 344 p. 8. Peter Jay Salzman, Michael Burian, Ori Pomerantz, Bob Mottram, Jim

	<p>Huang. The Linux Kernel Module Programming Guide. – 2021 (https://github.com/sysprog21/lkmpg/releases/download/latest/lkmpg.pdf)</p> <p>9. Pavel Yosifovich. Windows Kernel Programming. – Leanpub. – 2019. – 392 p.</p> <p>10. Rahul Sharma, Vesa Kaihlavirta, Claus Matzinger. The Complete Rust Programming Reference Guide, – Packt. – 2019. – 685 p.</p> <p>11. Steve Klabnik, Carol Nichols. The Rust Programming Language. – No Starch Press. – 2019. – 559 p.</p> <p>12. Andrew Weiss, Fullstack Rust. – Fullstack.io. – 2020. – 320 p.</p>
<p>Обсяг курсу</p>	<p>Загальний обсяг: 180 годин. Аудиторних занять: 64 год., з них 32 год. лекційних та 32 годин лабораторних робіт. Самостійної роботи: 116 год.</p>
<p>Очікувані результати навчання</p>	<p>Після завершення цього курсу студент буде:</p> <ul style="list-style-type: none"> - Знати: систематику технологій та основні тенденції в галузі системного програмування; теоретичні основи та сфери застосування системного програмування; основи програмування прикладних та системних задач із використанням обґрунтованих засобів. - Вміти: створювати програмні модулі для різних операційних платформ; застосовувати функції, що експортуються операційним середовищем; оптимізувати програмний код. <p>Після вивчення курсу «Системне програмування» здобувачі набудуть таких Загальних(ЗК)/Фахових(ФК) компетентностей та Програмних результатів навчання (ПРН):</p> <p>ЗК01. Здатність до абстрактного мислення, аналізу та синтезу.</p> <p>ЗК02. Здатність застосовувати знання у практичних ситуаціях.</p> <p>ЗК04. Здатність спілкуватися іноземною мовою як усно, так і письмово.</p> <p>ЗК05. Здатність вчитися і оволодівати сучасними знаннями.</p> <p>ЗК06. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.</p> <p>ФК14. Здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування.</p> <p>ФК15. Здатність розробляти архітектури, модулі та компоненти програмних систем.</p> <p>ФК17. Здатність дотримуватися специфікацій, стандартів, правил і рекомендацій в професійній галузі при реалізації процесів життєвого циклу.</p> <p>ФК18. Здатність аналізувати, вибирати і застосовувати методи і засоби для забезпечення інформаційної безпеки (в тому числі кібербезпеки).</p> <p>ФК23. Здатність реалізовувати фази та ітерації життєвого циклу програмних систем та інформаційних технологій на основі відповідних моделей і підходів розробки програмного забезпечення.</p> <p>ФК24. Здатність здійснювати процес інтеграції системи, застосовувати стандарти і процедури управління змінами для підтримки цілісності, загальної функціональності і надійності програмного забезпечення.</p> <p>ФК25. Здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення.</p> <p>ФК26. Здатність до алгоритмічного та логічного мислення.</p> <p>ПРН03. Знати основні процеси, фази та ітерації життєвого циклу програмного забезпечення.</p> <p>ПРН05. Знати і застосовувати відповідні математичні поняття, методи доменного, системного і об'єктно-орієнтованого аналізу та математичного моделювання для розробки програмного забезпечення.</p> <p>ПРН06. Вміння вибирати та використовувати відповідну задачу методо-</p>

	<p>логію створення програмного забезпечення.</p> <p>ПРН07. Знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення.</p> <p>ПРН11. Вибирати вихідні дані для проектування, куруючись формальними методами опису вимог та моделювання.</p> <p>ПРН12. Застосовувати на практиці ефективні підходи щодо проектування програмного забезпечення.</p> <p>ПРН13. Знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань.</p> <p>ПРН17. Вміти застосовувати методи компонентної розробки програмного забезпечення.</p> <p>ПРН21. Знати, аналізувати, вибирати, кваліфіковано застосовувати засоби забезпечення інформаційної безпеки (в тому числі кібербезпеки) і цілісності даних відповідно до розв'язуваних прикладних завдань та створюваних програмних систем.</p> <p>ПРН23. Вміти документувати та презентувати результати розробки програмного забезпечення.</p>
Ключові слова	Системне програмування, ядро ОС, програмні оболонки, системні мови програмування
Формат курсу	Очний
	Проведення лекцій, лабораторних робіт та консультації для кращого розуміння тем
Теми	Див. СХЕМА КУРСУ
Підсумковий контроль, форма	Залік у кінці семестру
Пререквізити	Для вивчення курсу студенти потребують базових знань з дисциплін «Програмування», «Архітектура комп'ютерів», «Алгоритми та структури даних», «Об'єктно-орієнтоване програмування».
Навчальні методи та техніки, які будуть використовуватися під час викладання курсу	Презентація, лекції, лабораторні роботи, обговорення, дискусія.
Необхідне обладнання	<p>Для проведення лекційних занять: комп'ютер (мінімальні характеристики: процесор з архітектурою x86_64, 8ГБ оперативної пам'яті, 512ГБ HDD/SSD), доступ до мережі Internet, засоби мультимедіа (в т.ч. проектор).</p> <p>Для проведення лабораторних занять: Комп'ютер (мінімальні характеристики: процесор з архітектурою x86_64, 8ГБ оперативної пам'яті, 512ГБ HDD/SSD). Необхідне програмне забезпечення включає в себе ОС Linux, ОС Windows evaluation edition, командні оболонки bash і Powershell, система контейнеризації Docker, компілятор та утиліти Rust.</p>
Критерії оцінювання (окремо для кожного виду навчальної діяльності)	<p>Оцінювання проводиться за 100-бальною шкалою. Бали нараховуються за наступним співвідношенням:</p> <ul style="list-style-type: none"> • Змістовий модуль 1: до 10 балів за виконання модульного завдання – написання есе за обраною тематикою з переліку. • Змістовий модуль 2: до 10 балів за виконання модульного завдання – написання есе за обраною тематикою з переліку. • Змістовий модуль 3: до 10 балів за виконання модульного завдання – написання есе за обраною тематикою з переліку. • Виконання лабораторних робіт: до 70 балів. <p>Підсумкова максимальна кількість балів 100. Звітність за курс – залік в кінці семестру.</p>

Змістовий модуль – самостійна робота студента оформлена у вигляді есе – одного із видів робіт, реалізованих у системі електронного навчання Moodle. Есе включає в себе детальний розгляд обраної індивідуальної теми, приведення прикладів та лістингів коду програм, огляду технологій, літератури. Код програм повинен бути обов'язково прокоментований та пояснений, необхідно також продемонструвати його роботу у разі, якщо в якості прикладу наводяться не окремі елементи технології, а суцільна програма. Есе також повинно містити посилання на літературні джерела/інтернет ресурси, що були використані під час його написання. Ілюстративні матеріали можна завантажувати у Moodle у вигляді файлів різних форматів. Теми для змістових модулів див. у розділі **Питання до модульного контролю**.

Академічна доброчесність: Очікується, що роботи студентів будуть оригінальними дослідженнями чи міркуваннями. Списування та втручання в роботу інших студентів становлять, але не обмежують, приклади можливої академічної недоброчесності. Виявлення ознак академічної недоброчесності в написанні завдань є підставою для її незарахування викладачем, незалежно від масштабів плагіату чи обману. Жодні форми порушення академічної доброчесності не толеруються.

Відвідання занять є важливою складовою навчання. Очікується, що всі студенти відвідають усі лекції та лабораторні зайняття курсу. Студенти повинні інформувати викладача про неможливість відвідати заняття. У будь-якому випадку студенти зобов'язані дотримуватися термінів визначених для виконання всіх видів робіт, передбачених курсом.

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.

Політика виставлення балів. Враховуються бали, набрані при поточному контролі та бали за виконання лабораторних робіт. При цьому обов'язково враховуються присутність на заняттях та активність студента під час практичного 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.

Оцінювання лабораторних робіт (10 лабораторних робіт, максимальна кількість балів: 70) відбувається шляхом оцінки роботи студента під час проведення лабораторної роботи в аудиторії та захисту звіту по виконаній лабораторній роботі (0-7 балів за одну роботу).

Бали оцінювання лабораторних робіт нараховуються за таким співвідношенням:

7 — робота виконана повністю та правильно, відповідає усім вимогам завдання. Код написаний грамотно, дотримано всіх правил стилю та форматування. Робота містить пояснювальні коментарі, які чітко описують принципи її роботи. Студент самостійно виконав роботу, без сторонньої допомоги.

6 — робота виконана повністю, але містить незначні помилки, які не впливають на її функціонування. Код написаний з деякими недоліками, але в цілому відповідає вимогам стилю та форматування. Робота містить пояснювальні коментарі, які достатньо описують принципи її ро-

боти. Студент виконав роботу самостійно, але потребував деякої допомоги викладача.

5 — робота виконана частково, але її функціональність достатня для виконання поставлених завдань. Код написаний з істотними недоліками, які впливають на читабельність та зрозумілість роботи. Робота містить пояснювальні коментарі, які недостатньо описують принципи її роботи. Студент виконав роботу самостійно, але потребував значної допомоги викладача.

4 — робота виконана частково, не виконує поставлених завдань. Код написаний з грубими помилками, які ускладнюють його читабельність та розуміння. Робота не містить пояснювальних коментарів. Студент виконав роботу з допомогою викладача.

3 — робота виконана частково. Код не написаний або написаний з численними грубими помилками, які унеможливають його виконання. Робота не містить пояснювальних коментарів. Студент не може повністю пояснити результати.

2 — робота виконана з використанням готових рішень, не містить власного програмного коду. Код не написаний або написаний з численними грубими помилками, які унеможливають виконання. Робота не містить пояснювальних коментарів. Студент не зміг пояснити, як працює його програма.

1 — робота не виконана, або виконана з використанням нечесних методів.

0 — відсутній звіт про виконання роботи.

Оцінювання змістових модулів (3 змістових модулів, 10 балів за кожний) — за результатами написаних студентом есе.

Бали оцінювання змістових модулів нараховуються за наступним співвідношенням:

9-10 — есе відповідає усім вимогам завдання, містить чітку та аргументовану відповідь на поставлене питання. В есе чітко викладені основні ідеї та думки, є логічні зв'язки між ними. Есе написане грамотно, дотримано всіх правил стилю та форматування. Есе містить оригінальні думки та ідеї, які розкривають тему есе з нового ракурсу.

7-8 — есе відповідає основним вимогам завдання, але містить деякі недоліки. В есе є чітка відповідь на поставлене питання, але вона не завжди аргументована. В есе є логічні зв'язки між ідеями, але вони не завжди чітко виражені. Есе написане грамотно, але має деякі недоліки у стилі та форматуванні. Есе містить оригінальні думки та ідеї, але вони не завжди розкриті повною мірою.

5-6 — есе відповідає загальним вимогам завдання, але має значні недоліки. В есе є відповідь на поставлене питання, але вона не завжди є аргументованою. В есе є логічні зв'язки між ідеями, але вони не завжди зрозумілі. Есе написане з деякими помилками у стилі. Есе містить оригінальні думки та ідеї, але вони не завжди є чіткими та зрозумілими.

3-4 — есе не відповідає основним вимогам завдання, має суттєві недоліки. В есе не завжди є чітка відповідь на поставлене питання, або відповідь є не аргументованою. В есе логічні зв'язки між ідеями не завжди зрозумілі або відсутні. Есе написане з численними помилками у стилі. Есе не містить оригінальних думок та ідей.

1-2 — есе не відповідає жодним вимогам завдання, має критичні недоліки. В есе немає чіткої відповіді на поставлене питання, або відповідь є не аргументованою. В есе логічні зв'язки між ідеями відсутні або є суперечливими. Есе написане з грубими помилками у стилі та формулюваннях. Есе не містить жодної оригінальної думки чи ідеї.

	<p>Критерії оцінювання результатів неформальної освіти: Нарахування балів відбувається за написання студентом тез доповідей на конференціях, наукових статей, участь у діяльності наукових гуртків, участь у наукових семінарах та круглих столах, конкурсах, участь у заходах неформальної освіти за отримання сертифікатів про проходження навчання на різних освітніх платформах (Coursera, Prometheus тощо), курсах на провідних ІТ компаніях за тематикою навчальної дисципліни. Кількість балів визначається відсотком покриття результатів відповідної активності до вимог результатів навчання з навчальної дисципліни.</p>
<p>Питання до контрольних робіт</p>	<p>Програмування в системних оболонках UNIX-сумісних ОС на прикладі Bash. Введення-виведення. Змінні. Розгалуження. Цикли. Функції. Шаблони відповідності. Регулярні вирази (РВ). Команди. Метасимволи РВ. Групування. Квадратні дужки і класи символів. Обернені посилання. Квантифікатори. Якорі та границі. Програмування в командній оболонці ОС Windows PowerShell. Змінні. Типи даних. Командлети. Об'єкти. Структури даних (масиви, ArrayList, хеш-таблиці). Потоки керування. Функції. Модулі. Регулярні вирази. Типові застосування PowerShell. Засоби деплойменту системного програмного забезпечення. Основи і задачі контейнеризації. Види контейнеризації і порівняння з віртуалізацією. Docker. Інсталяція Docker. Архітектура Docker. Формат Dockerfile. Базові образи. Мережева взаємодія. Під'єднання постійних сховищ. Публікація образів у реєстрі, Docker Hub. Модулі ядра Linux. Режими користувача та ядра. Конфігурація, компіляція та завантаження ядра Linux. Структура модуля ядра. Макроси <code>_init</code> та <code>_exit</code>. Параметри командного рядка. Функції, доступні для модулів. Операції введення-виведення. Драйвери символьних пристроїв. Структура <code>file_operations</code>. Реєстрація та дереєстрація пристроїв. Взаємодія із файловою системою <code>/proc</code>. Драйвери ядра Windows. Програмний інструментарій програмування драйверів Windows. Точка входу та вивантаження драйвера. Загальні правила програмування ядра Windows. Kernel API. Мови системного програмування. Порівняльний аналіз мов системного програмування. Реалізації менеджменту пам'яті, засоби автоматичного менеджменту пам'яті. Мова програмування Rust. Загальна характеристика та сфери застосування. Інструментарій програмування мовою Rust. Інсталяція Rust в різних операційних системах. Менеджмент проєктів за допомогою Cargo. Модулі та крейти, crates.io. Основні засоби та конструкції Rust. Типи даних, змінні, мутабельність, Функції. Замикання. Рядкові змінні. Розгалуження. Цикли. Визначені користувачем типи — структури й переліки. Методи типів. Колекції — масиви, кортежі, вектори, асоціативні масиви, зрізи. Ітератори. Дженерики і типажі (traits) у Rust Створення родових типів і функцій. Абстрактна поведінка на основі типажів. Форми типажів. Межі типажів. Стандартна бібліотека типажів. Менеджмент пам'яті в Rust. Концепція володіння. Области існування. Семантика переміщення та копіювання. Запозичення. Життєві цикли. Типи указників у Rust. Посилання. "Сирі" указники. Смарт-указники. Опрацювання помилок і конкурентність у Rust. Відновлювані та невідновлювані помилки. Комбінатори. Оператор ? Типаж Error. Підходи до паралельного виконання коду в Rust. Налаштовувані потоки. Доступ до даних з потоків. Arc і Mutex. RwLock. Синхронні і асинхронні канали. Огляд розширених можливостей Rust. Метапрограмування. macros_rules!. Процедурні та налагоджувальні макроси. Не безпечні функції та типажі. Інтерфейси до інших мов програмування. Логування.</p>
<p>Опитування</p>	<p>Анкету-оцінку з метою оцінювання якості курсу буде надано по завершенню курсу.</p>

**Схема курсу “Системне програмування”
для студентів спеціальності 121 – Інженерія програмного забезпечення**

Тиж.	Тема, план, короткі тези	Форма діяльності (заняття)	Література. Ресурси в Інтернеті	Завдання, год	Термін виконання
1,2	Програмування в системних оболонках UNIX-сумісних ОС на прикладі Bash. Введення-виведення. Змінні. Розгалуження. Цикли. Функції. Шаблони відповідності. Регулярні вирази (РВ). Команди. Метасимволи РВ. Групування. Квадратні дужки і класи символів. Обернені посилання. Квантифікатори. Якорі та границі.	Лекція	1, 2	4	кінець поточного тижня
1,2	Аналіз текстів засобами Bash	Лабораторна робота	Сайт курсу	4	кінець поточного тижня
3,4	Програмування в командній оболонці ОС Windows PowerShell. Змінні. Типи даних. Командлети. Об'єкти. Структури даних (масиви, ArrayList, хеш-таблиці). Потoki керування. Функції. Модулі. Регулярні вирази. Типові застосування PowerShell.	Лекція	3, 4	4	кінець поточного тижня
3,4	Аналіз текстів засобами PowerShell	Лабораторна робота	Сайт курсу	4	кінець поточного тижня
5,6	Засоби деплоюменту системного програмного забезпечення. Основи і задачі контейнеризації. Види контейнеризації і порівняння з віртуалізацією. Docker. Інсталяція Docker. Архітектура Docker. Формат Dockerfile. Базові образи. Мережева взаємодія. Під'єднання постійних сховищ. Публікація образів у реєстрі, Docker Hub.	Лекція	5, 6	4	кінець поточного тижня
5,6	Створення та публікація образів Docker	Лабораторна робота	Сайт курсу	4	кінець поточного тижня
7,8	Модулі ядра Linux. Режими користувача та ядра. Конфігурація, компіляція та завантаження ядра Linux. Структура модуля ядра. Макроси <code>_init</code> та <code>_exit</code> . Параметри командного рядка. Функції, доступні для модулів. Операції введення-виведення. Драйвери символічних пристроїв. Структура <code>file_operations</code> . Реєстрація та дереєстрація пристроїв. Взаємодія із файловою системою <code>/proc</code> .	Лекція	7, 8	4	кінець поточного тижня
7,8	Конфігурування, компіляція та завантаження ядра Linux	Лабораторна робота	Сайт курсу	4	кінець поточного тижня
9	Драйвери ядра Windows. Програмний інструментарій програмування драйверів Windows.	Лекція	9	2	кінець поточного тижня

	Точка входу та вивантаження драйвера. Загальні правила програмування ядра Windows. Kernel API.				
9	Модуль ядра — драйвер віртуального символьного пристрою	Лабораторна робота	Сайт курсу	2	кінець поточного тижня
10	Мови системного програмування. Порівняльний аналіз мов системного програмування. Реалізації менеджменту пам'яті, засоби автоматичного менеджменту пам'яті. Мова програмування Rust. Загальна характеристика та сфери застосування.	Лекція	10, 11, 12	2	кінець поточного тижня
10,11	Інструментарій програмування мовою Rust	Лабораторна робота	Сайт курсу	4	кінець поточного тижня
11	Інструментарій програмування мовою Rust. Інсталяція Rust в різних операційних системах. Менеджмент проєктів за допомогою Cargo. Модулі та крейти, crates.io.	Лекція	10, 11, 12	2	кінець поточного тижня
12	Основні засоби та конструкції Rust. Типи даних, змінні, мутабельність, Функції. Замикання. Рядкові змінні. Розгалуження. Цикли. Визначені користувачем типи — структури й переліки. Методи типів. Колекції — масиви, кортежі, вектори, асоціативні масиви, зрізи. Ітератори.	Лекція	10, 11, 12	2	кінець поточного тижня
12	Використання основних засобів мови Rust	Лабораторна робота	Сайт курсу	2	кінець поточного тижня
13	Дженерики і типажі (traits) у Rust Створення родових типів і функцій. Абстрактна поведінка на основі типажів. Форми типажів. Межі типажів. Стандартна бібліотека типажів.	Лекція	10, 11, 12	2	кінець поточного тижня
13	Крейти на основі бібліотеки функцій моделювання логічних елементів	Лабораторна робота	Сайт курсу	2	кінець поточного тижня
14	Менеджмент пам'яті в Rust. Концепція володіння. Области існування. Семантика переміщення та копіювання. Запозичення. Життєві цикли. Типи указників у Rust. Посилання. "Сирі" указники. Смарт-указники.	Лекція	10, 11, 12	2	кінець поточного тижня
14	Реалізація структур даних мовою Rust	Лабораторна робота	Сайт курсу	2	кінець поточного тижня
15	Опрацювання помилок і конкурентність у Rust. Відновлювані та невідновлювані помилки. Комбінатори. Оператор ? Типаж Error.	Лекція	10, 11, 12	2	кінець поточного тижня

	Підходи до паралельного виконання коду в Rust. Налаштовувані потоки. Доступ до даних з потоків. Arc і Mutex. RwLock. Синхронні і асинхронні канали.				
15,16	Створення розширень до інших мов програмування	Лабораторна робота	Сайт курсу	4	кінець поточного тижня
16	Огляд розширених можливостей Rust. Метапрограмування. macros_rules!. Процедурні та налагоджувальні макроси. Не безпечні функції та типажі. Інтерфейси до інших мов програмування. Логування.	Лекція	10, 11, 12	2	кінець поточного тижня