

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

**Затверджено**

На засіданні кафедри програмування  
факультету прикладної математики  
Львівського національного університету  
імені Івана Франка  
(протокол № 1 від 29 серпня 2024 р.)



Зав. кафедри к. ф.-м. н., доц. Ярошко С. А.

**Силабус навчальної дисципліни**  
**«Операційні системи та системне програмування»,**  
**викладається в межах ОПШ «01 Освіта/Педагогіка»**  
**першого (бакалаврського) рівня вищої освіти**  
**для здобувачів зі спеціальності 014.09 Середня освіта (Інформатика)**

Львів 2024 р.

<b>Назва дисципліни</b>	Операційні системи та системне програмування
<b>Адреса викладання дисципліни</b>	Львівський національний університет імені Івана Франка, вул. Університетська 1, м. Львів, Україна, 79000
<b>Факультет та кафедра, за якою закріплена дисципліна</b>	Факультет прикладної математики та інформатики, кафедра програмування
<b>Галузь знань, шифр та назва спеціальності</b>	Галузь знань: 01 Освіта/Педагогіка Спеціальність: 014.09 Середня освіта (Інформатика) Спеціалізація: Інформатика
<b>Викладачі дисципліни</b>	Черняхівський Володимир Вікторович, к. ф.-м. н., доц., доцент кафедри програмування
<b>Контактна інформація викладачів</b>	Електронна пошта: <a href="mailto:volodymyr.chernyakhivskyy@lnu.edu.ua">volodymyr.chernyakhivskyy@lnu.edu.ua</a> веб-сторінка: <a href="https://ami.lnu.edu.ua/employee/cherniakhivskiy">https://ami.lnu.edu.ua/employee/cherniakhivskiy</a>
<b>Консультації з питань навчання по дисципліні відбуваються</b>	Консультації проводять раз на тиждень згідно з оприлюдненим розкладом консультацій викладача. Можливі онлайн консультації через Zoom чи Microsoft Teams. Для погодження часу онлайн консультацій потрібно писати на електронну пошту викладача.
<b>Сторінка курсу</b>	<a href="https://ami.lnu.edu.ua/course/operatsiyni-systemy-ta-systemne-prohramuvannia-so">https://ami.lnu.edu.ua/course/operatsiyni-systemy-ta-systemne-prohramuvannia-so</a>
<b>Інформація про дисципліну</b>	Курс “Операційні системи та системне програмування” є вибірковою дисципліною зі спеціальності 014.09 Середня освіта (Інформатика) для освітньої програми «01 Освіта/Педагогіка», яку викладають у восьмому семестрі в обсязі 3 кредитів (за Європейською кредитно-трансферною системою ECTS)
<b>Коротка анотація дисципліни</b>	Вивчення елементів операційної системи, необхідних для оволодіння прийомами системного програмування. Формування системи знань про методи, алгоритми і прийоми програмування з використанням системних базових засобів комп'ютера. Формування практичних навичок складання системних програм шляхом моделювання реальних системних програм. Вивчення типових сучасних системних засобів програмування задач різного призначення, отримання знань і практичних навичок проектування системних програм, структур даних і побудови алгоритмічних моделей. Курс використовує низку знань і навичок, які були отримані студентами в інших раніше викладених курсах за розділами програмування і суміжних з ними.
<b>Мета та цілі дисципліни</b>	Метою нормативної дисципліни «Операційні системи та системне програмування» є: <ul style="list-style-type: none"> <li>● вивчення елементів операційної системи, пов'язаних з практичним програмуванням;</li> <li>● розуміння предмета дисципліни, зв'язку з іншими суміжними засобами і методами програмування, вивчення загальних принципів програмування задач на системному рівні операційної системи комп'ютера;</li> <li>● вивчення математичних та інформаційних основ системного програмування, як методу реалізації робочого середовища комп'ютера;</li> <li>● вміння застосовувати алгоритмічні мови асемблера, C++, C# та інші в обсязі задач розділів курсу;</li> <li>● вивчення сучасних алгоритмів програмування системних задач для типових застосувань в програмних проектах;</li> <li>● отримання навичок розробки процедур проектування, програмування, тестування і налагодження системних програм;</li> <li>● вміння використовувати стандартні базові бібліотеки операційної системи комп'ютера і середовищ програмування різних алгоритмічних мов.</li> </ul>

**Література для  
вивчення  
дисципліни**

*Основна література*

1. Шеховцов В.А. Операційні системи: підручник / В.А.Шеховцов // К.: Видавнича група ВНУ. – 2005. – 576с. – Режим доступу: <http://www.dut.edu.ua/ua/lib/1/category/729/view/1382>
2. Грайворонський М.В. Операційні системи: Методичні вказівки до комп'ютерного практикуму. [Електронний ресурс] : навч. посіб. для студ. спец. 113 «Прикладна математика», 125 «Кібербезпека» / КПІ ім. Ігоря Сікорського ; уклад.: М. В. Грайворонський, В. В. Демчинський – Електронні текстові дані (1 файл: 1,44 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2021. – 74 с. – Режим доступу: <https://ela.kpi.ua/server/api/core/bitstreams/990f4707-3aba-43b0-a930-b7e5d432211b/content>
3. Зайцев В.Г. Операційні системи. Конспект лекцій: [Електронний ресурс]: навч. посіб. для студ. спеціальності 123 «Комп'ютерна інженерія» / В. Г. Зайцев, І. П. Дробязко; КПІ ім. Ігоря Сікорського. – Електронні текстові дані (1 файл: 3 Мбайт). – Київ: КПІ ім. Ігоря Сікорського, 2019. – 240 с. – Режим доступу: <https://ela.kpi.ua/server/api/core/bitstreams/d9d83ff6-4004-4a8c-8772-62b624a2196b/content>
4. Галісеєв Г. Системне програмування / Геннадій Галісеєв // Університет "Україна". – 2019. – 113с.
5. Гаркуша І.М. Конспект лекцій з дисципліни “Операційні системи” для студентів галузі знань 12 “Інформаційні технології” спеціальності 126 “Інформаційні системи та технології”. – Д.: НТУ «ДП», 2020. – 73 с. – Режим доступу: [https://it.nmu.org.ua/ua/scientific\\_method\\_materials/lecture\\_notes/Конспект\\_лекцій\\_ОС\\_2020.pdf](https://it.nmu.org.ua/ua/scientific_method_materials/lecture_notes/Конспект_лекцій_ОС_2020.pdf)
6. Левченко Л.О. Операційні системи [Електронний ресурс] : навч. посіб. для здобувачів ступеня бакалавра за освіт. програмою «Цифрові технології в енергетиці» спец. 122 «Комп'ютерні науки» / Л. О. Левченко, Ю. А. Тарнавський ; КПІ ім. Ігоря Сікорського. – Електрон. текст. дані (1 файл. – Київ: КПІ ім. Ігоря Сікорського, 2023. – 256 с. – Режим доступу: <https://library.kre.dp.ua/Books/2-4%20kurs/Операційні%20системи/Levchenko-L-O-Operatsiini-systemy-KPI-2024-258s.pdf>
7. Порєв В.М. Системне програмування. Програмування на асемблері: комп'ютерний практикум [Електронний ресурс]: навч. посіб. для студ. освітньої програми «Комп'ютерні системи та мережі» спеціальності 123 «Комп'ютерна інженерія» / КПІ ім. Ігоря Сікорського; уклад. Порєв В.М. – Електронні текстові дані (1 файл: 3,2 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2022. – 146 с. – Режим доступу: <https://ela.kpi.ua/server/api/core/bitstreams/11d2c8c6-6b6d-4c2f-ab16-09a5d4b18fd6/content>
8. Портал знань. Асемблер (assembler) і системне програмування [Електронний ресурс]. – Режим доступу: <http://www.znannya.org/?view=asm>
9. Microsoft Ignite. What is a DLL [Електронний ресурс]. – Режим доступу: <https://support.microsoft.com/uk-ua/help/815065/what-is-a-dll>
10. Вікіпедія. Inkscape — вільний редактор векторної графіки [Електронний ресурс]. – Режим доступу: <https://uk.wikipedia.org/wiki/Inkscape>
11. Вікіпедія. Інтерпретатор мови програмування [Електронний ресурс]. – Режим доступу: <https://uk.wikipedia.org/wiki/Інтерпретатор>
12. Microsoft Ignite. Get Started with Win32 and C++ [Електронний ресурс]. – Режим доступу: <https://docs.microsoft.com/en-us/windows/win32/learnwin32/learn-to-program-for-windows>

*Додаткова література*

13. Інтерпретатор [Електронний ресурс]. – Режим доступу: <http://kytok.org.ua/?p=732>
14. Вікіпедія. Компілятор [Електронний ресурс]. – Режим доступу: <https://uk.wikipedia.org/wiki/Компілятор>

	<p>15. Microsoft Ignite. Visual Studio documentation [Електронний ресурс]. – Режим доступу: <a href="https://docs.microsoft.com/en-us/visualstudio/windows/?f1url=%3FappId%3DDev15IDEF1%261%3Den-US%26k%3Dk(MSDNSTART)%26rd%3Dtrue&amp;view=vs-2019">https://docs.microsoft.com/en-us/visualstudio/windows/?f1url=%3FappId%3DDev15IDEF1%261%3Den-US%26k%3Dk(MSDNSTART)%26rd%3Dtrue&amp;view=vs-2019</a></p> <p>16. Збірник електронних видань за курсом Системне програмування. Бібліотека Коледжу Радіоелектроніки [Електронний ресурс]. – Режим доступу: <a href="https://library.kre.dp.ua/Books/2-4%20kurs/Системне%20програмування/">https://library.kre.dp.ua/Books/2-4%20kurs/Системне%20програмування/</a></p>
<b>Обсяг курсу</b>	3 кредити ЄКТС – 90 годин. З них 28 годин лекцій, 28 годин лабораторних занять та 34 години самостійної роботи
<b>Очікувані результати навчання</b>	<p>Після завершення цього курсу студент буде:</p> <p><i>знати</i></p> <ul style="list-style-type: none"> <li>• принципи будови і функціонування сучасних операційних систем;</li> <li>• математичні та інформаційні основи системного програмування, як методу реалізації робочого середовища комп'ютера;</li> <li>• засоби алгоритмічних мов асемблера, С++, С# та інших в обсязі задач розділів курсу;</li> <li>• типові сучасні алгоритми програмування основних системних задач;</li> <li>• процедури проектування, програмування, тестування і налагодження системних програм;</li> <li>• стандартні базові бібліотеки операційної системи комп'ютера і середовищ програмування різних алгоритмічних мов.</li> </ul> <p><i>вміти</i></p> <ul style="list-style-type: none"> <li>• застосовувати засоби різних алгоритмічних мов для програмування системних задач;</li> <li>• складати, тестувати і налагоджувати системні програми загального і прикладного характеру;</li> <li>• аналізувати і модифікувати системні програми до зміни вимог;</li> <li>• реалізувати вимоги користувачів до будови робочого середовища;</li> <li>• застосовувати стандартні бібліотеки функцій операційної системи.</li> </ul>
<b>Компетентності</b>	<p><i>Інтегральна:</i> Здатність застосовувати загальні принципи програмування задач на системному рівні операційної системи комп'ютера мовами асемблера, С++, С#, та іншими. Здатність застосовувати сучасні системні засоби програмування задач різного призначення, проектувати системні програми, структури даних і алгоритмічні моделі.</p> <p><i>Загальні (ЗК):</i></p> <p>ЗК1. Здатність до абстрактного мислення, аналізу та синтезу, до застосування знань у практичних ситуаціях.</p> <p>ЗК2. Знання й розуміння предметної області та професійної діяльності.</p> <p>ЗК4. Здатність орієнтуватися в інформаційному просторі, здійснювати пошук, аналіз та обробку інформації з різних джерел, ефективно використовувати цифрові ресурси та технології в освітньому процесі.</p> <p><i>Спеціальні (фахові, предметні) компетентності (ФК та ПК):</i></p> <p>ФК1. Здатність перенесення системи наукових знань у професійну діяльність та в площину навчального предмету.</p> <p>ФК4. Здатність формувати і розвивати в учнів ключові та предметні компетентності засобами навчального предмету та інтегрованого навчання; формувати в них ціннісне ставлення, розвивати критичне мислення.</p> <p>ПК1. Здатність використовувати знання наукових фактів, концепцій, теорій, принципів і методів сучасної інформатики у практиці навчання інформатики.</p> <p>ПК3. Здатність до використання сучасних методів розробки та дослідження алгоритмів розв'язування задач у моделюванні об'єктів і процесів та реалізації цих алгоритмів сучасними мовами програмування.</p> <p>ПК4. Здатність використовувати програмні засоби загального та спеціального призначення для розв'язання прикладних задач з інформатики.</p>
<b>Програмні результати навчання</b>	ПРН7. Демонструє знання основ фундаментальних і прикладних наук інформатики та програмування, оперує базовими категоріями та поняттями предметної області спеціальності.

	<p>ПРН8. <i>Генерує</i> обґрунтовані думки в галузі професійних знань як для фахівців, так і для широкого загалу державною та іноземною мовами.</p> <p>ПРН9. <i>Застосовує</i> сучасні інформаційно-комунікаційні та цифрові технології у професійній діяльності.</p> <p>ПРН14. <i>Знає та розуміє</i> фізичні, логічні та математичні основи інформаційних технологій; пояснює та застосовує способи двійкового кодування текстової, числової, графічної, звукової та відеоінформації.</p>																																																											
<b>Ключові слова</b>	Операційна система, системне програмування, мікропроцесор, асемблер, компонування, зв'язування, об'єктний файл, параметри функцій, DLL, бібліотека функцій, комп'ютерна графіка, графічна операція, графічний редактор, електронна таблиця, сканер, лексичний аналіз, семантичний аналіз, інтерпретатор.																																																											
<b>Формат курсу</b>	Очний: проведення лекцій, лабораторних робіт та консультацій в приміщеннях університету, а в умовах форсмажорних обставин – онлайнний на платформі Microsoft Teams.																																																											
<b>Теми</b>	<table border="1"> <thead> <tr> <th>Тиж-день</th> <th>Тема, план, короткі тези</th> <th>Форма заняття</th> <th>Тривалість год</th> <th>Термін виконання</th> </tr> </thead> <tbody> <tr> <td rowspan="2">1</td> <td>Будова асемблерних програм. Схема трансляції, компонування і виконання програми. Основні реєстри мікропроцесора. Позиційна незалежність програми в однопрограмих ОС. Режими адресування (однопрограмих ОС). Моделі адресування. Моделі структури програм.</td> <td>Лекція</td> <td>2</td> <td></td> </tr> <tr> <td>Віконна програма перегляду і редагування текстових файлів. Файли Unicode і ASCII. Таблиці кодування символів.</td> <td>Лабораторна робота</td> <td>2</td> <td>Наступне лабораторне заняття</td> </tr> <tr> <td rowspan="2">2</td> <td>Бітова структура команд мікропроцесора. Загальний формат команди процесора Intel. Повна форма адресування операндів в пам'яті. Приклади трансляції команд в 32-бітовому режимі. Особливості адресування. Таблиці кодів команд та їх швидкодії.</td> <td>Лекція</td> <td>2</td> <td></td> </tr> <tr> <td>Програмний аналіз атрибутів файлів. Програмування сервісних засобів опрацювання файлів. Консольні і віконні реалізації програм. Процедури програмування, тестування і випуску програмного продукту.</td> <td>Лабораторна робота</td> <td>2</td> <td>Наступне лабораторне заняття</td> </tr> <tr> <td rowspan="2">3</td> <td>Трансляція і завантаження програм. Схеми виконання команд мікропроцесора. Комбінації операндів в командах. Форми запису констант в командах мовою асемблера. Директиви асемблера. Трансляція програм за першим і другим переглядом.</td> <td>Лекція</td> <td>2</td> <td></td> </tr> <tr> <td>Модель СОМ-програми. Зовнішня модель реалізації асемблера. Трансляція і виконання асемблерної програми. Перевірка моделі виконання СОМ-програми.</td> <td>Лабораторна робота</td> <td>2</td> <td>Наступне лабораторне заняття</td> </tr> <tr> <td rowspan="2">4</td> <td>Загальні принципи компонування програм. Об'єктні файли. Компонувальники і принципи їх роботи. Завантаження виконуваних файлів за статичного компонування.</td> <td>Лекція</td> <td>2</td> <td></td> </tr> <tr> <td>Трансляція ASM-команд. Схема трансляції. Підготовка даних до програмної реалізації. Визначення окремих асемблерних команд. Приклади для тестування асемблера.</td> <td>Лабораторна робота</td> <td>2</td> <td>Наступне лабораторне заняття</td> </tr> <tr> <td rowspan="2">5</td> <td>Динамічні бібліотеки. Загальні принципи будови. Переваги і недоліки використання динамічних бібліотек. Зворотня сумісність динамічних бібліотек. Неявне і явне зв'язування. Динамічні бібліотеки та адресний простір процесу. Особливості об'єктного коду динамічних бібліотек. Структура виконуваних файлів. Виконувані файли в Linux. Формат ELF. Виконувані файли у Windows. Формат PE. Динамічне компонування у Windows.</td> <td>Лекція</td> <td>2</td> <td></td> </tr> <tr> <td>Створення DLL. Будова бібліотеки DLL мовою С++. Проектування, програмування і тестування функцій DLL. Параметри функцій: типи Windows і стандартні типи С++. Компіляція функцій в бібліотеку DLL.</td> <td>Лабораторна робота</td> <td>2</td> <td>Наступне лабораторне заняття</td> </tr> <tr> <td>6</td> <td>Створення і використання функцій DLL. Стандартні типи даних Windows. Підготовка функцій для DLL. Компіляція функцій в бібліотеку DLL (створення DLL). Використання DLL. Неявне зв'язування, явне зв'язування.</td> <td>Лекція</td> <td>2</td> <td></td> </tr> </tbody> </table>					Тиж-день	Тема, план, короткі тези	Форма заняття	Тривалість год	Термін виконання	1	Будова асемблерних програм. Схема трансляції, компонування і виконання програми. Основні реєстри мікропроцесора. Позиційна незалежність програми в однопрограмих ОС. Режими адресування (однопрограмих ОС). Моделі адресування. Моделі структури програм.	Лекція	2		Віконна програма перегляду і редагування текстових файлів. Файли Unicode і ASCII. Таблиці кодування символів.	Лабораторна робота	2	Наступне лабораторне заняття	2	Бітова структура команд мікропроцесора. Загальний формат команди процесора Intel. Повна форма адресування операндів в пам'яті. Приклади трансляції команд в 32-бітовому режимі. Особливості адресування. Таблиці кодів команд та їх швидкодії.	Лекція	2		Програмний аналіз атрибутів файлів. Програмування сервісних засобів опрацювання файлів. Консольні і віконні реалізації програм. Процедури програмування, тестування і випуску програмного продукту.	Лабораторна робота	2	Наступне лабораторне заняття	3	Трансляція і завантаження програм. Схеми виконання команд мікропроцесора. Комбінації операндів в командах. Форми запису констант в командах мовою асемблера. Директиви асемблера. Трансляція програм за першим і другим переглядом.	Лекція	2		Модель СОМ-програми. Зовнішня модель реалізації асемблера. Трансляція і виконання асемблерної програми. Перевірка моделі виконання СОМ-програми.	Лабораторна робота	2	Наступне лабораторне заняття	4	Загальні принципи компонування програм. Об'єктні файли. Компонувальники і принципи їх роботи. Завантаження виконуваних файлів за статичного компонування.	Лекція	2		Трансляція ASM-команд. Схема трансляції. Підготовка даних до програмної реалізації. Визначення окремих асемблерних команд. Приклади для тестування асемблера.	Лабораторна робота	2	Наступне лабораторне заняття	5	Динамічні бібліотеки. Загальні принципи будови. Переваги і недоліки використання динамічних бібліотек. Зворотня сумісність динамічних бібліотек. Неявне і явне зв'язування. Динамічні бібліотеки та адресний простір процесу. Особливості об'єктного коду динамічних бібліотек. Структура виконуваних файлів. Виконувані файли в Linux. Формат ELF. Виконувані файли у Windows. Формат PE. Динамічне компонування у Windows.	Лекція	2		Створення DLL. Будова бібліотеки DLL мовою С++. Проектування, програмування і тестування функцій DLL. Параметри функцій: типи Windows і стандартні типи С++. Компіляція функцій в бібліотеку DLL.	Лабораторна робота	2	Наступне лабораторне заняття	6	Створення і використання функцій DLL. Стандартні типи даних Windows. Підготовка функцій для DLL. Компіляція функцій в бібліотеку DLL (створення DLL). Використання DLL. Неявне зв'язування, явне зв'язування.	Лекція	2	
Тиж-день	Тема, план, короткі тези	Форма заняття	Тривалість год	Термін виконання																																																								
1	Будова асемблерних програм. Схема трансляції, компонування і виконання програми. Основні реєстри мікропроцесора. Позиційна незалежність програми в однопрограмих ОС. Режими адресування (однопрограмих ОС). Моделі адресування. Моделі структури програм.	Лекція	2																																																									
	Віконна програма перегляду і редагування текстових файлів. Файли Unicode і ASCII. Таблиці кодування символів.	Лабораторна робота	2	Наступне лабораторне заняття																																																								
2	Бітова структура команд мікропроцесора. Загальний формат команди процесора Intel. Повна форма адресування операндів в пам'яті. Приклади трансляції команд в 32-бітовому режимі. Особливості адресування. Таблиці кодів команд та їх швидкодії.	Лекція	2																																																									
	Програмний аналіз атрибутів файлів. Програмування сервісних засобів опрацювання файлів. Консольні і віконні реалізації програм. Процедури програмування, тестування і випуску програмного продукту.	Лабораторна робота	2	Наступне лабораторне заняття																																																								
3	Трансляція і завантаження програм. Схеми виконання команд мікропроцесора. Комбінації операндів в командах. Форми запису констант в командах мовою асемблера. Директиви асемблера. Трансляція програм за першим і другим переглядом.	Лекція	2																																																									
	Модель СОМ-програми. Зовнішня модель реалізації асемблера. Трансляція і виконання асемблерної програми. Перевірка моделі виконання СОМ-програми.	Лабораторна робота	2	Наступне лабораторне заняття																																																								
4	Загальні принципи компонування програм. Об'єктні файли. Компонувальники і принципи їх роботи. Завантаження виконуваних файлів за статичного компонування.	Лекція	2																																																									
	Трансляція ASM-команд. Схема трансляції. Підготовка даних до програмної реалізації. Визначення окремих асемблерних команд. Приклади для тестування асемблера.	Лабораторна робота	2	Наступне лабораторне заняття																																																								
5	Динамічні бібліотеки. Загальні принципи будови. Переваги і недоліки використання динамічних бібліотек. Зворотня сумісність динамічних бібліотек. Неявне і явне зв'язування. Динамічні бібліотеки та адресний простір процесу. Особливості об'єктного коду динамічних бібліотек. Структура виконуваних файлів. Виконувані файли в Linux. Формат ELF. Виконувані файли у Windows. Формат PE. Динамічне компонування у Windows.	Лекція	2																																																									
	Створення DLL. Будова бібліотеки DLL мовою С++. Проектування, програмування і тестування функцій DLL. Параметри функцій: типи Windows і стандартні типи С++. Компіляція функцій в бібліотеку DLL.	Лабораторна робота	2	Наступне лабораторне заняття																																																								
6	Створення і використання функцій DLL. Стандартні типи даних Windows. Підготовка функцій для DLL. Компіляція функцій в бібліотеку DLL (створення DLL). Використання DLL. Неявне зв'язування, явне зв'язування.	Лекція	2																																																									

	Використання бібліотеки DLL в програмах C++ методами неявного і явного зв'язування. Файл імпорту, файл бібліотеки. Тестування прикладної програми, яка використовує DLL.	Лабораторна робота	2	Наступне лабораторне заняття
7	Методи застосування DLL. Загальна оцінка типів даних функцій DLL. Функції DLL в програмах C#. Функції високої швидкодії і точності. Зовнішні функції в Python. Типи параметрів зовнішніх функцій Python. Функції DLL в програмах мовою асемблера. Явне і неявне зв'язування з DLL мовою асемблера.	Лекція	2	
	Використання DLL мовами програмування. Функції DLL в програмах C# (неявне зв'язування з бібліотеками DLL). Функції DLL в програмах компільованих алгоритмічних мов. Ефективне використання DLL мовами інтерпретованого типу (Python, Visual Basic, Java).	Лабораторна робота	2	Наступне лабораторне заняття
8	Принципи програмування графіки в операційній системі. Відображення малюнків на екрані дисплея. Векторна і растрова графіка. Спрощена масштабована векторна графіка в JavaScript. Формати графічних файлів. Перетворення графічних форматів.	Лекція	2	
	Базові елементи програмування графіки. Підготовка основи для програмування графічних операцій. Стандартні класи бібліотеки, які забезпечують програмне малювання. Методи реалізації графічних операцій: векторна графіка, растрова графіка. Реалізація дигамічного малювання окремих фігур.	Лабораторна робота	2	Наступне лабораторне заняття
9	Алгоритми виконання графічних операцій. Параметри стандартного графічного інструментарія. Схема будови програмованих класів малювання. Приклад будови активних елементів на час виконання.	Лекція	2	
	Алгоритми і програмна реалізація окремих графічних операцій – параметри і сервісні можливості. Скасування операцій. Копіювання і пересування окремих фігур чи ділянок малюнка.	Лабораторна робота	2	Наступне лабораторне заняття
10	Методи і технології програмування графічних операцій. Інтерфейсна частина проєкта. Меню вікон верхнього рівня. Вікна з меню і з панеллю інструментів. Підхід до функцій опрацювання графіки. Графіка на основі пакетів бібліотечних функцій типових технологій (PIL, Pillow, numpy, інші).	Лекція	2	
	Додаткові графічні операції. Перегляд і редагування великих за розмірами малюнків. Програмування операцій фільтрування. Програмування операцій обертання малюнка.	Лабораторна робота	2	Наступне лабораторне заняття
11	Обчислення формул інтерпретацією. Різновиди інтерпретаторів. Проміжний код. Правила будови синтаксичних визначень. Інтерпретація формул загальних алгебраїчних правил. Приклад програмної реалізації. Розширення змісту формул. Операторне виконання.	Лекція	2	
	Модель інтерпретатора формул для операцій без рангів. Доповнення інтерпретатора бінарними і унарними операціями. Вибір системи числення. Опрацювання помилок в формулах.	Лабораторна робота	2	Наступне лабораторне заняття
12	Побудова сканера для формул загального виду. Структури даних сканера. Граматичні правила для чисел і формул. Лексеми. Формати лексем і алгоритми розпізнавання. Визначення класу сканера.	Лекція	2	
	Модель інтерпретатора формул за алгебраїчними правилами. Бінарні і унарні операції. Граматика виразів. Метод рекурсивного спуску для граматичного розбору. Розширення змісту формул. Доповнення інтерпретатора.	Лабораторна робота	2	Наступне лабораторне заняття
13	Перетворення виразів з інфіксної форми в постфіксну. Граматичні правила для формул. Програмування перетворювача.	Лекція	2	
	Програмування і застосування сканера формул формату електронної таблиці. Формати лексем і алгоритми розпізнавання. Діаграми станів і скінчені автомати. Перетворення в постфіксну форму. Обчислювач формул.	Лабораторна робота	2	Наступне лабораторне заняття
14	Програмні проєкти за розділами системного програмування. Вибір алгоритмічної мови і середовища реалізації. Проєктування і програмування з використанням стандартних класів і бібліотек. Інкапсульовані стандартні функції операційної системи. Параметри оцінки проєктів.	Лекція	2	
	Аналіз проєктів, виконаних за курсом ОСіСП. Виправлення і доопрацювання окремих функцій.	Лабораторна робота	2	Початок сесії

	Остаточне налагодження інтерфейсних частин проєктів. Підготовка до іспиту.
<b>Підсумковий контроль, форма</b>	Іспит в кінці семестру. Перелік питань до іспиту і порядок проведення будуть оголошені до початку екзаменаційної сесії.
<b>Пререквізити</b>	Для вивчення курсу студенти потребують базових знань з дисциплін: дискретна математика, архітектура обчислювальних систем та комп'ютерна схемотехніка, програмування, алгоритми і структури даних, програмна інженерія.
<b>Навчальні методи та техніки, які використовують під час викладання курсу</b>	Лекції з мультимедійними презентаціями та з демонстрацією прийомів програмування системних засобів; лабораторні заняття у вигляді проєктування алгоритмів і програм, виконання практичних завдань та проєктів; самостійне опрацювання навчальних матеріалів: підручників, конспектів лекцій, електронних ресурсів, готових програм різними алгоритмічними мовами, додаткових матеріалів, розміщених у хмарному сховищі (Microsoft Teams, Google Диск). Обговорення теоретичного та практичного матеріалу в онлайн сервісах, формулювання творчих завдань для студентів, виконання яких готує до вивчення нового теоретичного і практичного матеріалу.
<b>Необхідне обладнання</b>	Для проведення лекцій: комп'ютер, проєктор, доступ до мережі інтернет. Для проведення лабораторних та виконання завдань: комп'ютер; ОС Windows/Linux; доступ до інтернету; середовища програмування мовою C++ (Microsoft Visual Studio, Code Blocks тощо), мовою C#, мовою Python (IDLE Python тощо), мовою асемблера, мовою Java. Можливі використання інших мов програмування. Уся література і робочі матеріали, які 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.
<b>Критерії оцінювання (окремо для кожного виду навчальної діяльності)</b>	<b>Оцінювання</b> проводиться за 100-бальною шкалою. 50 балів нараховують за виконання лабораторних завдань та індивідуальних проєктів впродовж семестру (поточний контроль), ще 50 балів – за виконання екзаменаційного завдання.  <b>Оцінювання завдань і проєктів</b> (поточний контроль). Курс складається з чотирьох розділів. За кожним з чотирьох розділів студент має виконати за своїм вибором або лабораторні роботи, або реалізувати програмний проєкт (системну програму). Лабораторні роботи індивідуальні. Проєкти можуть бути індивідуальні та командні. Упродовж семестру студент виконує до 12 лабораторних робіт і до 4 проєктів (у власній комбінації, лабораторні чи проєкти є альтернативними), які оцінюють різною шкалою залежно від складності, шкала оцінювання і критерії надаються студентам на початку семестру. Для кожного лабораторного завдання визначено термін виконання – зазвичай до наступного лабораторного заняття. Для проєктів час на виконання збільшений, його визначають окремою шкалою.  <b>Параметри оцінювання програм:</b> загальна працездатність; відсутність помилок виконання і побічних ефектів; незалежність від інших програм; функціональна повнота реалізації завдання; виконання команд через меню і клавіатуру (клавіатурний інтерфейс); наявність функцій, побудованих за власними алгоритмами; можливість відміни операцій – там, де є потенційна потреба; коректна робота з дуже великими і дуже малими об'єктами; коректність відкриття, закриття та інших операцій з файлами; оперативні підказки в процесі роботи; контроль за коректністю команд користувача; наявність довідкової інформації про функціональні можливості програми.  Вчасно виконані завдання поточного контролю оцінюють так (у відсотках від максимальної оцінки): <ul style="list-style-type: none"> <li>• 100% – умови завдання виконано повністю, алгоритми складено правильно, програма містить належні коментарі, роботу програми перевірено на достатньому наборі тестових даних, автор відповідає на всі запитання щодо використаних підходів, чітко інтерпретує отримані результати, немає ознак недоброчесності;</li> </ul>

- 80% – наведено логічно правильну послідовність розв’язування, алгоритми складено правильно, бракує окремих коментарів чи тестів, автор не досить повно пояснює використані підходи, немає ознак недоброчесності;
- 60% – у правильній послідовності розв’язування допущено окремі помилки, які автор уміє виправити після зауваження викладача, бракує коментарів чи тестів, на запитання щодо використаних підходів автор відповідає з помилками, немає ознак недоброчесності;
- 40% – у правильній послідовності розв’язування пропущено окремі етапи, завдання виконано частково, автор не розуміє недоліків поданої роботи, не вміє їх виправити, немає ознак недоброчесності;
- 20% – завдання виконано частково, немає тестів, програма працює правильно для окремих наборів вхідних даних, автор не може самостійно інтерпретувати отримані результати, виправити помилки, немає ознак недоброчесності;
- 0% – завдання не виконано, написана програма не відповідає умові, або ж виявлено ознаки недоброчесності: запозичення, фрагменти коду, дію яких автор пояснити не може, автор не володіє відповідним теоретичним матеріалом тощо;
- можуть бути нараховані додаткові бали за повністю виконане завдання, яке містить кілька способів розв’язування, використовує особливо ефективний спосіб, демонструє креативність автора тощо.

Запізнення терміну виконання завдання зменшує максимальну оцінку за завдання, ступінь зменшення визначає викладач до кожної роботи.

**Екзаменаційний контроль.** Іспит в кінці семестру. Загальний список питань до екзамену надається окремо на початку семестру. Екзаменаційний білет складається з чотирьох питань відповідно до розділів курсу. У відповіді до всіх питань, крім текстових викладів, записати і пояснити необхідні фрагменти програмних кодів і структур даних. Їх можна взяти з власних раніше виконаних проєктів чи лабораторних робіт. Програмні коди і структури даних вважаються підтвердженням змісту текстового викладу. Незалежно від використаних матеріалів має бути власна частина текстового викладу як розуміння змісту чи особиста інтерпретація, тобто власна точка зору до змісту питання.

Оцінювання відповідей:

- за питання розділу "асемблер" 0-14 балів;
- за питання розділів "Бібліотеки DLL", "Графічний редактор", "Електронні таблиці та інтерпретатори" 0-12 балів кожне.

Разом 0-50 балів.

Оцінка за екзаменаційне завдання може отримати додаткові бали, якщо проєкти студента чи окремі лабораторні роботи, виконані впродовж семестру, мали закінчений характер і високий рівень фахової реалізації.

**Відвідання занять** є важливою складовою навчання. Очікується, що всі студенти відвідають усі лекції і лабораторні заняття курсу. Активність під час проведення лекцій і лабораторних заохочується балами при оцінюванні відповідного лабораторного завдання чи проєкту. У будь-якому випадку студенти зобов’язані дотримуватися усіх строків визначених для виконання усіх видів робіт, передбачених курсом. Виконані роботи завантажують у відповідне хмарне сховище. Альтернативою відвідування лабораторних занять в університеті може бути дистанційна онлайн робота за розкладом проведення занять.

**Академічна доброчесність:** очікується, що роботи студентів будуть їхнім оригінальними дослідженнями чи міркуваннями. Відсутність посилань на використані джерела, фабрикування джерел, списування, втручання в роботу інших студентів, здавання чужих комп’ютерних програм як своїх 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.

**Опитування**

Анкету-оцінку з метою оцінювання якості курсу буде надано після завершення курсу.



