

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

Затверджено

На засіданні кафедри радіофізики та комп'ютерних технологій факультету електроніки та комп'ютерних технологій Львівського національного університету імені Івана Франка
(протокол № ____ від _____ червня 2021 р.)

Завідувач кафедри _____

Силабус з навчальної дисципліни
«Крос-платформне програмування»,
що викладається в межах ОПП «Комп'ютерні науки» першого
(бакалаврського) рівня вищої освіти для здобувачів зі
спеціальності
122 «Комп'ютерні науки»

Назва дисципліни	Крос-платформне програмування
Адреса викладання дисципліни	м. Львів, вул. Тарнавського, 107
Факультет та кафедра, за якою закріплена дисципліна	Факультет електроніки та комп'ютерних технологій, кафедра радіофізики і комп'ютерних технологій
Галузь знань, шифр та назва спеціальності	12 Інформаційні технології, 122 Комп'ютерні науки
Викладачі дисципліни	Хвищун Іван Олександрович , кандидат технічних наук, доцент
Контактна інформація викладачів	Ivan.Khvyshchun@lnu.edu.ua , xiocompan@gmail.com
Консультації з питань навчання по дисципліні відбуваються	Консультації проводяться в день проведення лекційних занять (за попередньою домовленістю) згідно розкладу. В режимі он-лайн: консультації проводяться за домовленістю з студентами чи групою студентів на платформі MS Teams та на інших електронних ресурсах, за попереднім погодженням часу (через електронну пошту викладача або телефонний дзвінок).
Сторінка дисципліни	
Інформація про дисципліну	Дисципліна «Крос-платформне програмування» є нормативною дисципліною циклу професійної та практичної підготовки зі спеціальності 122 Комп'ютерні науки для освітньої програми «Комп'ютерні науки», яка викладається в 3 семестрі в обсязі 4-х кредитів (за Європейською Кредитно-Трансферною Системою ECTS).
Коротка анотація дисципліни	Навчальна дисципліна «Крос-платформне програмування» забезпечує набуття студентами спеціальності знань та умінь для створення крос-платформних програмних продуктів на базі опанування відповідних технологій.
Мета та цілі дисципліни	Мета: одержання студентами базових знань стосовно сучасних методологій розробки програмного забезпечення в крос-платформній парадигмі програмування, зокрема – вивчення основ програмування алгоритмічними мовами C# та Java, вивчення типів даних та алгоритмічних конструкцій цих мов. Цілі: навчити студентів методології розробки крос-платформних програм, розуміти синтаксичні та семантичні конструкції алгоритмічних мов C# та Java, отримати навички розробки та налагодження програм у консольних та віконних режимах роботи програмних середовищ Visual Studio 2019 та Java Beans, а також, опанувати принципи розробки крос-платформних проєктів для розв'язання найпростіших задач фізики та електроніки.
Література для вивчення дисципліни	Основна література: <ol style="list-style-type: none"> 1. Троелсен Э., Джепикс Ф. Язык программирования C# 7 и платформы .NET и .NET Core, 8-е изд.: – СПб.: ООО “Диалектика”, 2018 – 1328 с. 2. Троелсен Э. Язык программирования C# 2010 и платформа .NET 4.0, 5-е изд. ООО “И.Д. Вильямс”, 2011. – 1392 с. 3. Нэш Т. C# 2010: ускоренный курс для профессионалов. Пер. с англ. – М.: ООО “И.Д. Вильямс”, 2010. – 592 с. 4. Рихтер Дж. CLR via C#. Программирование на платформе Microsoft .NET Framework 2.0 на языке C#. Мастер-класс. – 2-е изд. – М.: Изд. «Русская Редакция»; СПб.: Питер, 2008. – 656 с. 5. C# Language Specification Version 4.0 1999-2010 Microsoft Corporation – Стандарт мови C#. 6. Рысованый А.Н. Системное программирование, Ч.1. Программирование в среде masm64: учеб.-метод. пособие – Харьков: «Слово», 2017. – 108 с.

Обсяг курсу	64 години аудиторних занять. З них 32 години лекцій, 32 години лабораторних робіт на кожен із 8 бригад та 56 годин самостійної роботи
Очікувані результати навчання	В результаті вивчення даного курсу студент повинен: знати: <ul style="list-style-type: none"> • принципи технології розробки крос-платформних програмних систем; • принципи використання засобів крос-платформного програмування; • теоретичні основи у галузі застосування засобів крос-платформного програмування. вміти: <ul style="list-style-type: none"> • використовувати базові принципи створення крос-платформних програмних систем для розроблення відповідних проектів; • застосовувати теоретичні знання та практичні навички у галузі крос-платформного програмування; • опанувати сучасні методи та технології створення платформо-незалежного програмного забезпечення.
Ключові слова	Архітектура. Платформа. Мова програмування. Середовище програмування. Клас. Делегат. Інтерфейс. База даних.
Формат курсу	Очний
	Проведення лекцій, лабораторних робіт та консультації для кращого розуміння тем
Теми	Див. СХЕМА КУРСУ
Підсумковий контроль, форма	Один колоквиум впродовж семестру у письмовій формі та іспит в кінці семестру в усній та практичній формі
Пререквізити	Для вивчення курсу студенти потребують базових знань з дисциплін «Алгоритмізація та програмування», «Об'єктно-орієнтоване програмування», «Операційні системи».
Навчальні методи та техніки, які будуть використовуватися під час викладання курсу	Лекції на основі презентацій, лабораторні роботи на основі методичок та їхній захист на основі індивідуальних звітів, виконання індивідуальних завдань та їхній захист.
Необхідне обладнання	Пристрої мультимедіа. ОС Windows 10, Visual Studio 2019, платформи MS Teams, Moodle.
Критерії оцінювання (окремо для кожного виду навчальної діяльності)	Оцінювання проводиться упродовж семестру за 100-бальною шкалою. Бали нараховуються за такими видами робіт з наступним співвідношенням: <ul style="list-style-type: none"> • лабораторні роботи + індивідуальні завдання: 40% семестрової оцінки; максимальна кількість балів 40. • контрольні заміри (1 модуль): 10% семестрової оцінки; максимальна кількість балів 10. • іспит: 50% семестрової оцінки, максимальна кількість балів - 50 Загалом, упродовж семестру: 100 балів. <p>Контрольні заміри проводяться у формі письмового колоквиуму та поточного захисту індивідуальних звітів по кожній лабораторній та індивідуальній роботі.</p> Академічна доброчесність: Очікується, що лабораторні індивідуальні роботи студентів будуть їхніми оригінальними розробками та дослідженнями. Відсутність посилань на використані джерела, фабрикування джерел, списування, втручання в роботу інших студентів становлять, але не обмежують, приклади можливої академічної не доброчесності. Виявлення ознак академічної не доброчесності в роботі студента є підставою для її не зарахування викладачем, незалежно від

	<p>масштабів плагіату чи обману.</p> <p>Відвідання занять є важливою складовою навчання. Очікується, що всі студенти відвідають усі лекції та виконають усі основні та додаткові завдання до кожної лабораторної роботи. Студенти повинні інформувати викладача про неможливість відвідати заняття. Студенти зобов'язані дотримуватися усіх термінів визначених для виконання усіх видів робіт, передбачених курсом.</p> <p>Література. Основна та додаткова література надається викладачем в електронній формі виключно в освітніх цілях без права її передачі третім особам. Студенти заохочуються до пошуку та використання іншої літератури та джерел по тематиці курсу, яких немає серед рекомендованих.</p> <p>Політика виставлення балів. Враховуються бали набрані при захисті лабораторних та індивідуальних робіт, бали за колоквиуми та 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 та ін.</p> <p>Жодні форми порушення академічної доброчесності не толеруються.</p>
Питання до контрольних робіт	Питання до колоквиуму студенти отримують за тиждень до дати колоквиуму. Білети для підсумкового іспиту студенти отримують на останній лекції курсу.
Опитування	Анкету-оцінку з метою оцінювання якості курсу буде надано по завершенню курсу.

СХЕМА КУРСУ

Тиждень	Тема, план, короткі тези	Форма діяльності (заняття)	Література. Ресурси в Інтернеті	Завдання (лабораторна робота), год	Термін виконання
1	<p>АРХІТЕКТУРА INTEL-СУМІСНОГО МП. ВСТУП В АСЕМБЛЕР</p> <p>Мета та задачі курсу. Правила виконання лабораторного практикуму. Принципи та способи поточного і підсумкового контролю успішності. Література основна та допоміжна.</p> <p>Крос-платформість рівня мікропроцесора і рівня операційної системи. Поняття архітектури ЕОМ. Архітектура фон-нейманівського комп'ютера. Що таке Асемблер. Етапи розробки автономної програми на Асемблері. Формат команд Асемблера. Регістри загального призначення. Функціональне призначення сегментів програми. Типи даних в Асемблері. Моделі пам'яті. Система команд Intel-сумісних МП. Система переривань. Математичний копроцесор.</p>	Лекція	1, 2, 3, 4, 5, 6	Робота у 16-ти розрядному Асемблері.	1 тиждень семестру

2	<p>КЛАСИФІКАЦІЯ МП. 32-х-та 64-х РОЗРЯДНИЙ АСЕМБЛЕР. ВСТУП У .NET FRAMEWORK</p> <p>Поняття операційної системи. Класифікаційні ознаки МП. Мікропроцесорні платформи (архітектури). Переваги та недоліки Асемблера. Захищений режим роботи МП. Програмування на Асемблері задач захищеного режиму. Компілювання MASM. Поняття Microsoft .NET. Огляд .NET Native. Сучасний стан платформи .NET. Складові частини .NET Framework.</p>	Лекція	1, 2, 3, 4, 5, 6	Робота у 32-х розрядному Асемблері.	2 тиждень семестру
3	<p>МОЖЛИВОСТІ ПЛАТФОРМИ .NET Framework. Вступні зауваження. Історія розвитку платформи .NET Framework. Складові частини .NET Framework. Середовище Mono. Xamarin. Сьогоднішні варіанти платформи .NET від Microsoft. Огляд .Net Core. Складові частини .NET Core. Visual Studio Code. Поняття про рефакторинг.</p>	Лекція	1, 2, 3, 4, 5, 6	Робота в автономному компіляторі C#	3 тиждень семестру
4	<p>ВСТУП У ПРОГРАМУВАННЯ МОВОЮ C#. Алфавіт мови C#. Коментарі. Змінні. Ключові слова. Літерали (константи). Типи даних мови C#. Спільна система типів даних (CTS) мови C#. Вбудовані типи даних. Суфікси у літералах цілочисельних типів даних. Символьні типи. Суфікси у літералах дійсних типів даних.</p>	Лекція	1, 2, 3, 4, 5, 6	Робота у консольному режимі роботи VS C# 2019	4 тиждень семестру
5	<p>ОПЕРАТОРИ. СТРУКТУРОВАНІ ТИПИ ДАНИХ</p> <p>Оператори мови C#. Оператор присвоєння. Блок, або складений оператор. Оператори розгалуження. Оператори безумовного переходу. Оператори умовних переходів (розгалужень): оператор if (скорочений варіант), оператор if (повний варіант), оператор вибору switch...case.</p> <p>Оператори циклів: цикл з відомою кількістю повторень for, цикл з передумовою while, цикл з після умовою do...while, оператори дострокового завершення циклу, оператор foreach – цикл по контейнеру.</p> <p>Структуровані типи даних. Масиви: двовимірні масиви, ініціалізація двовимірного масиву, тривимірні масиви, зубчаті (jagged) масиви. Класи. Структури. Перелічування.</p>		1, 2, 3, 4, 5, 6	Віконний проект МДН_МН у VS C# 2019	5 тиждень семестру

6	РЕАЛІЗАЦІЯ ТЕХНОЛОГІЇ ООП У C# Част. 1. Поняття об'єктної моделі. Принципи побудови об'єктної моделі. Поля класу. Методи класу: параметри методів, статичні методи	Лекція	1, 2, 3, 4, 5, 6	Програмування методів розв'язання СЛАР у C#	6 тиждень семестру
7	РЕАЛІЗАЦІЯ ТЕХНОЛОГІЇ ООП У C# Част. 2. Перевантаження методів. Конструктори. Конструктори та ініціалізатори об'єктів. Оголошення власних конструкторів. Статичний конструктор. Деструктори. Синтаксис оголошення деструктора. Властивості. Модифікатори доступу властивостей. Автоматичні властивості. Скорочений запис властивості	Лекція	1, 2, 3, 4, 5, 6	Метод Ньютона для розв'язування СНР	7 тиждень семестру
8	ІНКАПСУЛЯЦІЯ. УСПАДКУВАННЯ. ІНДЕКСАТОРИ. Інкапсуляція. Успадкування і поліморфізм. Успадкування по принципу належності (відношення «має»). Віртуальні методи. Пізніше і раннє зв'язування об'єктів класу. Абстрактні класи. Приховування членів класу. Перетворення типів поміж базовими і дочірніми класами. Ключові слова is і as. Індексатори. Ключове слово this.	Лекція	1, 2, 3, 4, 5, 6	Вивчення елементів ООП у мові C#	8 тиждень семестру
9	CALLBACK-ФУНКЦІЇ. ПОДІЇ. ДЕЛЕГАТИ. Поняття події. Події та їхня реалізація подій з погляду Delphi. Делегати. Опрацювання подій.	Лекція	1, 2, 3, 4, 5, 6	Вивчення поняття Делегат	9 тиждень семестру
10	ПОДІЇ ТА АНОНІМНІ МЕТОДИ. ЛЯМБДА-ВИРАЗИ. Деталізація поняття події. Анонімні методи. Лямбда-вирази. Зразки лямбда-виразів. Запитання для перевірки засвоєння матеріалу.	Лекція	1, 2, 3, 4, 5, 6	Вивчення поняття Подія	10 тиждень семестру
11	ПОБУДОВА ГРАФІЧНИХ ЗОБРАЖЕНЬ ЗАСОБАМИ GDI+. Вступне зауваження. Елементи GDI і GDI+. Простори імен GDI+. Контексти пристроїв і об'єкт Graphics. Малювання контурів. Виведення двох графіків на компоненту pictureBox. Математичний та програмний аспекти побудови графіків. Використання нашої програми для ілюстрування МНК. Апроксимування методом найменших квадратів (МНК): лінійний та нелінійний випадки	Лекція	1, 2, 3, 4, 5, 6	Вивчення графіки C# на основі методу найменших квадратів	11 тиждень семестру
12	АТРИБУТИ. ІНТЕРФЕЙСИ. Вступні зауваження. Атрибути. Обмеження на застосування атрибутів. Інтерфейси. Приклад опису інтерфейсу. Приклад	Лекція	1, 2, 3, 4, 5, 6	Розробка демо-проекту "Інтерфейси"	12 тиждень семестру

	інтерфейсу із різними елементами. Приклад множинного успадкування інтерфейсів. Приклад реалізації інтерфейсів. Важливі типові інтерфейси. Інтерфейс ICloneable. Інтерфейси IEnumerable, IEnumerator.				
13	НАЛАГОДЖЕННЯ ПРОГРАМ. ОБРОБКА ВИКЛЮЧЕНЬ. ФАЙЛИ. Розроблення надійного програмного забезпечення (ПЗ). Налагодження програм. Режими Debug та Release. Вікно виведення Output. Точки трасування і точки зупинки. Виняткові ситуації (exerption). Оператори try – catch. Робота з файловою системою. Робота з дисками. Робота з файлами із використанням класу FileStream. Інший спосіб створення об'єкта. Властивості і методи класу FileStream. Читання та записування файлів.	Лекція	1, 2, 3, 4, 5, 6	Розробка і проекту Sklad	13 тиждень семестру
14	СТВОРЕННЯ НАВЧАЛЬНОГО ПРОЕКТУ ДЛЯ РОБОТИ ІЗ НАБОРАМИ ДАНИХ. ЕЛЕМЕНТИ ТЕХНОЛОГІЇ БАЗ ДАНИХ. Доступ до даних засобами платформи .NET. Розробка візуального інтерфейсу проекту Sklad. Послідовність кроків для реалізації проекту. Програмування функціоналу проекту. Формування підсумків про наявний товар. Програмування спадного меню проекту. Програмування довідників Під'єднання довідника до кольорової панелі.	Лекція	1, 2, 3, 4, 5, 6	Удосконалення проекту Sklad	14 тиждень семестру
15	ЕЛЕМЕНТИ ТЕХНОЛОГІЇ БАЗ ДАНИХ (БД). Основні поняття теорії БД. Класифікація БД. Моделі БД. Властивості БД. Системи керування БД (СКБД). Класифікація СКБД. Вступ в ADO.NET. Елементи мови T-SQL. Основні поняття мови інтегрованих запитів LINQ. Варіант LINQ to Objects. Різновиди синтаксису мови LINQ. Основи Entity Framework. Концепції Entity Framework. Способи взаємодії з БД. Підхід Entity Framework Code First. Взаємодія з даними. Підхід Code First до існуючої БД. Використання ключів. Підхід Database First. Підхід Model First.	Лекція	1, 2, 3, 4, 5, 6	Створення БД та її під'єднання до проекту Sklad	15 тиждень семестру
16	ЕЛЕМЕНТИ МОВИ JAVA. Вступ у Java Beans. Графічний інтерфейс користувача у Java Beans. Правила для класів у Java Beans.	Лекція		Створення проекту мовою Java	16 тиждень семестру

	Створення компонентів JavaBeans. Файл маніфесту. Типи даних, оператори та керуючі конструкції мови Java. Реалізація принципів ООП у мові Java. Масиви. Контейнери. Система введення/виведення у Java				
--	--	--	--	--	--