

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЛЬВІВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
імені ІВАНА ФРАНКА**

Кафедра безпеки інформації та бізнес-комунікацій

Затверджено

на засіданні кафедри безпеки інформації та
бізнес-комунікацій
економічного факультету
Львівського національного університету імені
Івана Франка
(протокол №6 від 11.01.2023 р.)

В.о. зав. кафедри  проф. М.І. Хмельничук

СИЛАБУС З НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

**«НАВЧАЛЬНА ПРАКТИКА»,
що викладається в межах ОІІІ**

«Економічна кібернетика та бізнес-аналітика»

**першого (бакалаврського) рівня вищої освіти для здобувачів зі
спеціальностей : 051 «Економіка»**

Львів - 2023 р.

Назва курсу	“ Навчальна практика”
Адреса викладання дисципліни	м. Львів, проспект В'ячеслава Чорновола, 61, аудиторія ____
Факультет та кафедра, за якою закріплена дисципліна	Економічний факультет, кафедра безпеки інформації та бізнес-комунікацій
Галузь знань, шифр та назва спеціальності	05 Соціальні та поведінкові науки 051 Економіка
Ступінь вищої освіти	Бакалавр
Статус дисципліни	вибіркова навчальна дисципліна
Семестр	6
Форма навчання	Денна
Обсяг дисципліни, кредити ЄКТС / загальна кількість годин	6 кредитів / 180 годин
Викладач (-і)	Циганчук Роман Олегович, кандидат економічних наук, доцент, доцент кафедри безпеки інформації та бізнес-комунікацій економічного факультету
Контактна інформація викладача (-ів)	Профайл викладача курсу: https://econom.lnu.edu.ua/employee/tsyhanchuk-roman-olehovych Електронна пошта roman.tsyhanchuk@lnu.edu.ua
Консультації з питань вивчення дисципліни	У день проведення практичних занять, 16.00-17.00 год. (м. Львів, проспект В'ячеслава Чорновола, 61, аудиторія 325)
Мова викладання	Українська
Сторінка курсу	

ІНФОРМАЦІЯ ПРО ДИСЦИПЛІНУ

Коротка анотація дисципліни	Силабус вибіркової навчальної дисципліни «Навчальна практика» складений відповідно до освітньо-професійної програми підготовки фахівця освітнього ступеня «бакалавр». Вивчення дисципліни «Навчальна практика» спрямоване на оволодіння знаннями розробки алгоритмів та програмування, що дозволить реалізовувати задачі автоматизації обробки інформації, автоматизації керування об'єктами, за допомогою комп'ютерної техніки. Такі знання майбутній спеціаліст зможе застосовувати як при подальшому навчанні, так і після отримання вищої освіти у своїй професійній діяльності.
Мета дисципліни	Мета вивчення дисципліни є ознайомлення студентів з основними принципами технології ООП та оволодіння ними основами мови програмування C++, методами проектування та створення програм згідно сучасних технологій програмування. Завданням дисципліни є оволодіння знаннями розробки алгоритмів та програмування, що дозволить реалізовувати задачі автоматизації обробки інформації, автоматизації керування об'єктами, в тому числі, економічними, за допомогою комп'ютерної техніки. Такі

	<p>знання майбутній спеціаліст зможе застосовувати як при подальшому навчанні, так і після отримання вищої освіти у своїй професійній діяльності.</p> <p>У результаті вивчення дисципліни студент повинен знати принципи ООП (інкапсуляція, спадкування, поліморфізм), як оголошуються класи, створюються, ініціалізуються і руйнуються об'єкти класів, відмінність між абстрактними і конкретними класами, ієрархію класів потоків введення-виведення.</p> <p>Після вивчення дисципліни студент повинен уміти створювати класи і об'єкти, управляти доступом до даних і функцій, ініціалізувати об'єкти за допомогою конструкторів, використовувати спадкування і віртуальні функції для поліморфізму, створювати, читати і записувати файли, формувати дані, що виводяться на екран, а також використовувати свої знання й уміння в курсовому і дипломному проектуванні.</p>
<p>Література для вивчення дисципліни</p>	<p style="text-align: center;">Базова:</p> <ol style="list-style-type: none"> 1. Кузнецов М.С. Об'єктно-орієнтоване програмування з використанням UML та мови C++: Навч. посібник. – Дніпропетровськ: НМетАУ, 2003. – 90 с. 2. Ludwig R. Verbesserubg einer Iterationsfolge bei Gleichungssystemen, Z. Angew. Manh. Mech, 2011, 232-234. 3. 7. Luxemburg W. On tye convergence of successive approximations in the theory of ordinary differential equations I-III: (I) Canad. Math. Bull., 9-20; (II) Nederl. Acad. Wetensch. Proc., ser. A 61 (indag. Math.), 20, 540-546; (III) Nieuw. Arch. Wisk. (3), 6, 93-98, 2009. 4. 8. Shinbrot M. A fixed point theorem and some applications, Arch. Rational. Mech. Anal., 17 255-271. 5. 9. Schmidt J., Schwetlick H. Ableitungsfreie Verfahren mit hoherer Konvergenzgeschwindigkeit, Computing, 3, 2009, 215-226. 6. 10. Schwartz J. Nonlinear functional analysis, Lecture Notes, Courant inst. Of Math. Sci., New York Univ, 1991. 7. Крис Паппас, Уильям Мюррей. Программирование на C и C++. Серия «Библиотека студента». – «Ирина», ВНУ, Киев, 2000. – 320 с. 8. ANSI, American National Standart for Information Systems – Programming Language C. – New York, 1990. 9. Бублик, Володимир Васильович. Об'єктно-орієнтоване програмування : підручник для студентів, які навчаються за напрямками "Комп'ютерні науки", "Комп'ютерна інженерія", "Програмна інженерія", "Системна інженерія", "Інформатика", "Прикладна математика" / В.В. Бублик. - Київ : ІТ-книги, 2015. - 637 с. 10. Алхімова С. М. Алгоритмізація та програмування. Програмування мовою високого рівня C++ : методичні вказівки до виконання лабораторних робіт. - Київ : НТУУ "КПІ", 2015. - 105 с. 11. Щербаков О.В. Основи об'єктно-орієнтованого програмування : навчальний посібник / О. В. Щербаков, Ю. Е. Парфьонов, В. М. Федорченко. – Харків : Вид. ХНЕУ ім. С. Кузнеця, 2019. – 246 с. 12. Коноваленко І.В. Програмування мовою C# 7.0 : навчальний посібник / Коноваленко І.В.,– Тернопіль:Тернопільський національний технічний університет імені Івана Пулюя 2017 –

	300 с.
Обсяг курсу	64 годин аудиторних занять, з них 64 годин практичних занять та 116 годин самостійної роботи
Очікувані результати навчання	<p>У результаті вивчення дисципліни студент повинен знати принципи ООП (інкапсуляція, спадкування, поліморфізм), як оголошуються класи, створюються, ініціалізуються і руйнуються об'єкти класів, відмінність між абстрактними і конкретними класами, ієрархію класів потоків введення-виведення.</p> <p>Після вивчення дисципліни студент повинен уміти створювати класи і об'єкти, управляти доступом до даних і функцій, ініціалізувати об'єкти за допомогою конструкторів, використовувати спадкування і віртуальні функції для поліморфізму, створювати, читати і записувати файли, форматовувати дані, що виводяться на екран, а також використовувати свої знання й уміння в курсовому і дипломному проектуванні.</p>
Ключові слова	інкапсуляція, спадкування, поліморфізм, клас, конструктор, ініціалізація.
Формат курсу	Денний Проведення практичних занять, консультацій
Теми	<p>Тема 1. Основні поняття об'єктно-орієнтованого програмування.</p> <p>Тема 2. Класи мовою С++ та об'єкти.</p> <p>Тема 3. Конструктори і деструктори.</p> <p>Тема 4. Спадкування.</p> <p>Тема 5. Поліморфізм.</p> <p>Тема 6. Абстрактні класи.</p> <p>Тема 7. Класи вводу-виводу та Файлові потоки вводу-виводу в С++.</p>
Підсумковий контроль, форма	<p>Критерії оцінювання</p> <p>1. Критерієм успішного проходження здобувачем освіти оцінювання може бути досягнення ним мінімальних порогових рівнів оцінок за кожним запланованим результатом навчання навчальної дисципліни.</p> <p>2. Мінімальний пороговий рівень оцінки варто визначати за допомогою якісних критеріїв і трансформувати його в мінімальну позитивну оцінку числової (рейтингової) шкали, що використовується.</p> <p>Засоби оцінювання</p> <p>Засобами оцінювання результатів навчання можуть бути:</p> <ul style="list-style-type: none"> – стандартизовані тести; – аналітичні звіти, реферати, есе; – розрахункові та розрахунково-графічні роботи; презентації результатів виконаних завдань та досліджень; розрахункові роботи; – інші види індивідуальних та групових завдань. <p>Форми поточного та підсумкового контролю</p> <ol style="list-style-type: none"> 1. Форма підсумкового контролю за навчальною дисципліною «Навчальна практика» - залік. 2. Форми поточного контролю під час навчальних занять: усні відповіді. Розв'язування задач та практичних завдань, письмове опитування у формі самостійних та контрольних робіт,

	написання економічних есе.	
Пререквізити	<p>Вивчення дисципліни «Навчальна практика» спрямоване на оволодіння знаннями розробки алгоритмів та програмування, що дозволить реалізовувати задачі автоматизації обробки інформації, автоматизації керування об'єктами, за допомогою комп'ютерної техніки. Такі знання майбутній спеціаліст зможе застосовувати як при подальшому навчанні, так і після отримання вищої освіти у своїй професійній діяльності.</p> <p>Міждисциплінарні зв'язки: дисципліна «Навчальна практика» ґрунтується на знаннях, отриманих при вивченні таких курсів як «Економічний аналіз», «Цифрова економіка», «Вища математика», «Інформаційні технології (рівень А)» та інших курсів.</p>	
Навчальні методи та техніки, які будуть використовуватися під час викладання курсу	<p>Практичні заняття (навчальні дискусії, мозковий штурм, розв'язок ситуаційних вправ (кейсів)); Самостійне навчання (індивідуальна робота, робота в групах).</p> <p>Практичні заняття сконструйовані з застосуванням методів практико-орієнтованого навчання, і передбачають розв'язок здобувачами вищої освіти кейсів на основі можливих реальних ситуацій та виконання необхідних розрахунків. Самостійне навчання сприяє підготовці до лекцій, практичних занять, а також роботи індивідуально та в невеликих групах для підготовки презентацій, що будуть представлені іншим групам, та для виконання індивідуальних та групових ситуаційних вправ на практичних заняттях, участі в них тощо.</p>	
Необхідні обладнання	<p>Мультимедіа та проекційна апаратура. Комп'ютери, комп'ютерні системи та мережі. Бібліотечні фонди.</p>	
Критерії оцінювання (окремо для кожного виду навчальної діяльності)		Максимальна кількість балів
	Форми контролю	Денна форма навчання
	Поточний контроль:	
	Усні відповіді, розв'язування задач та практичних завдань	10 x 5 = 50 балів
	Письмові опитування	до 30 балів
	Підсумкова контрольна робота	20 балів
	Всього за результатами поточного контролю:	100
	Всього	100
<p>Письмові роботи: Очікується, що студенти виконають декілька видів письмових робіт (есе, вирішення кейсу).</p> <p>Академічна доброчесність: Очікується, що роботи студентів будуть їх оригінальними дослідженнями чи міркуваннями. Відсутність посилань на використані джерела, фабрикавання 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>Відвідування занять є важливою складовою навчання. Очікується, що всі студенти відвідають усі лекції і практичні заняття курсу. Студенти мають інформувати викладача про неможливість</p>		

	<p>відвідати заняття. У будь-якому випадку студенти зобов'язані дотримуватися усіх строків визначених для виконання усіх видів письмових робіт, передбачених курсом.</p> <p>Жодні форми порушення академічної доброчесності <u>не толеруються</u>.</p>
<p>Контрольні запитання</p>	<ol style="list-style-type: none"> 1. Які переваги використання символічної константи замість літеральної? 2. Яка різниця між <code>if(x=3)</code> та <code>if(x==3)</code>? 3. В чому різниця між оголошенням прототипу функції та визначенням функції? 4. Чи повинні імена параметрів, що вказані в прототипі, визначенні та виклику функції відповідати одні одним? 5. Що таке локальна змінна? 6. Що таке область видимості? 7. Що таке рекурсія? 8. Коли варто використовувати глобальні змінні? 9. Що таке перевантаження функції? 10. Що таке поліморфізм? 11. Що таке оператор прямого доступу і для чого він використовується? 12. Що резервує пам'ять оголошення чи визначення? 13. Оголошення класу є його інтерфейсом чи виконанням? 14. Яка різниця між відкритими (<code>public</code>) та закритими (<code>private</code>) даними-членами класу? 15. Чи можуть функції-члени бути закритими? 16. Чи можуть змінні-члени бути відкритими? 17. Яка функція викликається при ініціалізації класу? 18. Як здійснюється обмеження доступу до членів класу? 19. Коли викликаються конструктор і деструктор? 20. Які види циклів ви знаєте? 21. Чи можна в циклі <code>for</code> ініціалізувати зразу декілька змінних-лічильників? 22. Наведіть приклад циклу, який ніколи не завершиться. 23. Що відбувається при запуску безкінечного циклу? 24. В чому різниця між циклами <code>while</code> і <code>do...while</code>? 25. Що таке вказівник? 26. В чому перевага використання вказівників? 27. Який оператор використовується для отримання адреси змінної? 28. Який оператор дозволяє отримати значення записане за адресою? 29. В чому переваги посилань над вказівниками? 30. В чому переваги вказівників над посиланнями? 31. Коли варто повертати результат функції як значення? 32. В чому різниця між вказівником і посиланням? 33. Що таке посилання? 34. Як оголосити посилання? 35. Якщо ви перевантажили функцію-член, як можна відрізнити різні варіанти функції? 36. Яка різниця між визначенням та оголошенням? 37. Коли викликається конструктор-копіювальник? 38. Як відрізняється перевантаження операторів префіксної та постфіксної дії? 39. Як перевантажити оператори? 40. Для чого здійснюється перевантаження операторів? 41. Що таке усадкування?

	<p>42. Що таке віртуальний деструктор?</p> <p>43. Чи може бути віртуальний конструктор?</p> <p>44. Для чого використовуються віртуальні функції? Наведіть приклад.</p> <p>45. З якою метою використовується ключове слово protected?</p> <p>46. Що таке масив?</p> <p>47. Що буде, якщо записати дані за межі масиву?</p> <p>48. Що таке поліморфізм?</p> <p>49. Що таке абстрактний тип?</p> <p>50. Як оголосити чисту віртуальну функцію?</p> <p>51. Для чого використовуються абстрактні типи даних?</p> <p>52. Назвати основні компоненти бази даних і пояснити функцію кожної з них.</p> <p>53. Назвати основні моделі даних та дати їх характеристику.</p>
Опитування	<p>Анкету-оцінку з метою оцінювання якості курсу буде надано по завершенню курсу.</p>