Факультет інформаційних технологій
Кафедра цифрової економіки та системного аналізу

СИЛАБУС (SYLLABUS)
Дисципліна «Інструментальні засоби прикладного програмування/ Application programming tools»

ІНФОРМАЦІЯ ПРО ВИКЛАДАЧА

Викладач	Міценко Сергій Анатолійович
Науковий ступінь	Кандидат технічних наук
Вчене звання	Доцент
Посада	Доцент кафедри цифрової економіки та системного аналізу
Адреса кафедри	м.Київ, вул. Кіото 19, каб. Б-517, Б-519
E-mail	desa@knute.edu.ua
Консультації	Відповідно до графіку індивідуальних консультацій на сайті кафедри

ПОЛІТИКА АКАДЕМІЧНОЇ ДОБРОЧЕСНОСТІ

https://knute.edu.ua/file/MzEyMQ==/c12a9f74e87d9154696ca0f761da2e5c.pdf

Дотримання академічної доброчесності передбачає:

- самостійне виконання навчальних завдань, завдань поточного та підсумкового контролю результатів навчання (для осіб з особливими освітніми потребами ця вимога застосовується з урахуванням їхніх індивідуальних потреб і можливостей);
- посилання на джерела інформації у разі використання не авторських ідей, розробок, тверджень, відомостей і т.п.;
- дотримання норм законодавства про авторське право і суміжні права;
- надання достовірної інформації про результати власної наукової діяльності, використані методики досліджень і джерела інформації.

Порушенням академічної доброчесності вважається:

- академічний плагіат – оприлюднення (частково або повністю) наукових (творчих) результатів, отриманих іншими особами, як результатів власного дослідження (творчості) та/або відтворення опублікованих текстів (оприлюднених творів мистецтва) інших авторів без зазначення авторства;
- самоплагіат – оприлюднення (частково або повністю) власних раніше опублікованих наукових результатів як нових наукових результатів;
- фабрикація – вигадування даних чи фактів, що використовуються в наукових дослідженнях;
- фальсифікація – свідома зміна чи модифікація вже наявних даних, що стосуються наукових досліджень.

За порушення академічної доброчесності здобувачі освіти можуть бути притягнені до академічної відповідальності:

- повторне проходження оцінювання (модульний контроль, іспит, залік тощо);
- повторне проходження відповідного освітнього компонента освітньо-професійної програми;
- відрахування з Університету;
- позбавлення наданих університетом пільг;

- відмова у присудженні відповідного ступеня вищої освіти;

ПОЛІТИКА ЩОДО ВІДВІДУВАННЯ ЗАНЯТЬ

- відвідування занять є обов'язковим;
- Студент, який пропустив практичне заняття, самостійно вивчає матеріал (при виникненні питань може звертатися за консультацією згідно розкладу консультацій викладачів оприлюдненого на сайті кафедри) за наведеними джерелами, виконує завдання і здає його викладачу.
- за об'єктивних причин (наприклад, хвороба, міжнародне стажування та ін.) навчання може відбуватись в он-лайн формі за погодженням із викладачем дисципліни.

ОПИС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Назва дисципліни / тип дисципліни	Інструментальні засоби прикладного програмування / обов'язкова
Навчальний рік	2022-2023
Факультет	Факультет інформаційних технологій
Курс	2
Семестр	4
Освітній ступінь	Молодший бакалавр
Галузь знань	12 «Інформаційні технології»
Спеціальність	122 «Комп'ютерні науки»
Загальна характеристика	Кількість годин –180 Кількість кредитів – 6 **Види занять:** лекції, лабораторні, самостійна робота. **Співвідношення аудиторних годин і годин самостійної роботи** - 102/78 **Мова викладання** – українська **Форма викладання** – очна
Підсумковий контроль	Екзамен
Програмне забезпечення	C, C++
Обладнання	Проектор, комп'ютерна техніка із встановленим програмним забезпеченням та доступом до мережі Інтернет.
Необхідні попередні дисципліни	«Комп'ютерні технології обробки та візуалізації даних»
Методика вивчення	Методика вивчення дисципліни полягає у набутті студентами знань теоретичного і практично-прикладного характеру під час лекцій, лабораторних занять, самостійної роботи та вивчення першоджерел і навчально-методичної літератури.
Мета і завлання	**Метою** вивчення дисципліни "Інструментальні засоби прикладного програмування" є формування у студентів навичок оволодіння технологіями обробки простих та структурованих даних, опанування технологій структурного, модульного та об'єктно-орієнтованого програмування на базі мови програмування C++. **Завданням** вивчення дисципліни є теоретична та практична підготовка студентів з таких питань: • парадигми програмування (структурного, модульного, об'єктно-орієнтованого) та засоби сучасних мов програмування для реалізації різних концепцій; • засоби мов програмування для реалізація розгалужених та цикліч-

	них алгоритмів; • використання функцій та окремих модулів користувача; • реалізація класичних інформаційних структур (списків, дерев) з використанням статичного та динамічного розподілу пам'яті; • базові технології проектування процесів пошуку та побудови впорядкованих даних у лінійних списках та деревовидних структурах; • використання класів для реалізації парадигми об'єктно-орієнтованого програмування; • засоби створення ієрархічної об'єктної структури з використанням базового об'єкта, подальше розширення його властивостей з використанням статичних та віртуальних методів; • сучасні інструментальні засоби для створення прикладного програмного забезпечення. Автоматизація програмування: технологія RAD (Rappid Application Development); • особливості програмування в операційних системах родини Windows. Система управління повідомленнями. Структура програми для Windows; • програмування графічного інтерфейсу користувача, ресурси прикладної програми. Функції API (Application Programming Interface); • застосування бібліотеки класів MFC (Microsoft Foundation Classes) як інструменту для створення прикладних програм для Windows; • специфікація COM (Component Object Model) як основа технології створення компонент ActiveX. Поняття інтерфейсу компоненти, полімор-фізм, таблиця віртуальних функцій, множинні екземпляри; • динамічне компонування прикладної програми.
Місце дисципліни в освітньо-професійній програмі	
Загальні компетентності	ЗК 2 Здатність застосовувати знання у практичних ситуаціях. ЗК 3 Знання та розуміння предметної області та розуміння професійної діяльності. ЗК 4 Здатність спілкуватися державною мовою як усно, так і письмово в термінології предметної області. ЗК 7 Здатність до пошуку, оброблення та аналізу інформації з різних джерел.
Фахові компетентності (результати навчання)	СК 6 Здатність застосовувати методології, технології та інструментальні засоби для розробки програмних систем, продуктів і сервісів інформаційних технологій. СК 8 Здатність проектувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: узагальненого, об'єктно-орієнтованого, функціонального, логічного, з відповідними моделями, методами й алгоритмами обчислень, структурами даних і механізмами управління.
Програмні результати навчання	ПР 4 Використовувати високорівневі мови програмування, обчислювальні методи і алгоритми для розробки програмних засобів обробки даних. ПР 8 Використовувати інструментальні засоби проектування концептуальних, логічних та фізичних моделей баз даних, створювати бази даних, розробляти та оптимізувати запити до них, у тому числі із застосуванням мов програмування.

ТЕМАТИКА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

РОЗДІЛ 1. Основи програмування за допомогою Application Programming Interface (API).

Тема 1.1. Знайомство з середовищами розроблення програм

Види програмного забезпечення для розроблення програм в операційній системі Windows. Знайомство з інструментальними засобами MS Visual Studio. Майстри для створення проектів різних типів. Поняття проекту в MS Visual C++. Основні структурні компоненти проекту. Аналіз програмного забезпечення, що створюється майстрами MS Visual C++. Компіляція та запуск програм на виконання. Інтерактивні засоби Microsoft Visual C++ для налагодження програм.

Тема 1.2. Основи програмування в операційній системі Windows

Структура програми Application Programming Interface (API) для Windows. Функція WinMain. Вікно як основний об'єкт програми для Windows. Клас вікна та його реєстрація. Типи вікон. Створення вікна програми. Цикл оброблення повідомлень. Механізм передачі повідомлень. Структура віконної процедури.

Тема 1.3. Керування виведенням інформації в системі Windows

Контекст пристрою. Види графічних об'єктів. Режими відображення та різні системи координат. Робота з текстом. Функції API для керування графічним виводом.

РОЗДІЛ 2. ЗАСОБИ ДЛЯ РОЗРОБЛЕННЯ ПРОГРАМ У СИСТЕМАХ WINDOWS З ЗАСТОСУВАННЯМ БІБЛІОТЕКИ КЛАСІВ MFC.

Тема 2.1. Створення MFC-програм

Призначення бібліотеки класів MFC (Microsoft Foundation Classes). Аналіз класів для програм з одним та декількома документами. Поліморфізм та віртуальні функції, перевантаження віртуальних функцій в успадкованих класах. Призначення та структура карти повідомлень (message maps). Макроси, що використовуються в карті повідомлень. Створення та редагування карти повідомлень за допомогою майстра. Створення функцій для оброблення повідомлень Windows (Windows message handler). Поняття команди (command). Створення функцій для оброблення команди.

Тема 2.2. Створення графічного інтерфейсу користувача в MFC-програмах

Формування ресурсів вікна діалогу. Створення класу вікна діалогу. Обмін даними між програмою та вікном діалогу за допомогою змінних класу. Робота з елементами управління різних типів. Формування ресурсів меню. Створення функції для оброблення команди меню. Формування ресурсів панелей інструментів. Створення функції для оброблення команди кнопки панелі інструментів.

Тема 2.3. Виведення даних у вікно документа в MFC-програмах

Призначення класу CPaintDC. Клас відображення CView. Використання функцій класу для виведення інформації документа. Робота з графічними об'єктами в MFC-програмах. Графічні примітиви. Призначення полів структури Windows LOGFONT. Використання олівців (pens) та пензликів (brushes).

Тема 2.4. Створення додатку на основі діалогових вікон

Призначення класу додатку CApp, структура файлу оголошення класу та файлу функцій. Призначення та реалізація класів відображення CDialog. Перевантаження функцій та створення повідомлень і функцій обробки повідомлень. Створення власних класів на основі класів MFC. Застосування класів CMetaFileDC збереження документу при роботі додатку.

Тема 2.5. Створення додатку на основі концепції DocView

Призначення класу додатку CApp, класів CMainFrame, CDoc та CView структура файлу оголошення класів та файлу функцій. Реалізація класів відображення CDialog та CView. Перевантаження функцій та створення повідомлень і функцій обробки повідомлень. Створення власних класів на основі класів MFC. Застосування класів похідних вікон CChildFraim та імплементації контейнеру CCntrlItem. Застосування класу CDoc для збереження документу додатку.

Тема 2.6. Робота з базами даних у MFC-програмах

Сучасні технології доступу до реляційних баз даних. Технологія ODBC. Використання класу CDataBase для оброблення бази даних. Створення вікон діалогу для відображення записів таблиць бази даних. Використання класів CRecordset, CRecordView для роботи з записами таблиць бази даних. Використання SQL (Structured Query Language) для одержання інформації із декількох таблиць бази даних.

Тема 2.7. Основи моделі компонентних об'єктів Microsoft

Основні поняття технології ActiveX- Приклади. Модель компонентних об' єктів (COM - Component Object Model) як основа технології ActiveX. Поняття компоненти. Архітектура прикладної програми на основі компонент. Поняття інтерфейсу компоненти. Взаємодія з компонентою за допомогою її інтерфейсів. Призначення інтерфейсу IUnknown. Віртуальні функції інтерфейсу IUnknowh. Запит інтерфейсу компоненти за допомогою функції QuerylnterFace. Керування життям компоненти. Функції AddRef та Release.

Тема 2.8. Динамічне компонування програми

Поняття динамічної бібліотеки DLL (Dynamic Link Library). Переваги динамічного компонування програми. Експорт функції із DLL. Розбиття монолітної програми на файли клієнта та компоненти. Розміщення компоненти в динамічній бібліотеці.

Перелік навчальних робіт студентів та оцінки їх у балах з дисципліни «Інструментальні засоби прикладного програмування»

Види робіт	К-сть балів
Лабораторне заняття №1. Тема: «Знайомство з середовищами розроблення програм».	3
Лабораторне заняття №2. Тема: «Програмування рекурсивного алгоритму малювання у головному вікні додатку».	3
Лабораторне заняття №3. Тема: «Програмування із застосуванням стилізованих пер та пензлів».	3
Лабораторне заняття №4. Тема: «Створення MFC-програми на основі вікон діалогу».	3
Лабораторне заняття №5. Тема: «Використання елементів керування та ресур-сів, для створення інтерактивної програми діалогу».	3
Лабораторне заняття №6. Тема: «Побудова простої інтерактивної програми малювання із застосуванням MFC».	3
Лабораторне заняття №7. Тема: «Застосування елементів управління, контекс-ту пристрою та виклик системних діалогових вікон».	3
Лабораторне заняття №8. Тема: «Застосування функцій відображення графіч-ний примітивів та класу CMetaFileDC».	3
Лабораторне заняття №9. Тема: «Створення простої програми малювання на основі концепції DocView»	3
Лабораторне заняття №10. Тема: «Створення інтерфейсу малювання та за-собів збереження фігур графічних примітивів».	3
Лабораторне заняття №11. Тема: «Створення додаткового інтерфейсу перегляд великого документу та друк».	4
Лабораторне заняття №12. Тема: «Робота з БД у MFC програмах».	4
Лабораторне заняття №13. Тема: «Створення та застосування компонент засобами бібліотеки MFC»	4

Лабораторне заняття №14. Тема: «Розробка вікна діалогу як компоненти власного додатку».	4
Лабораторне заняття №15. Тема: «Створення динамічної бібліотеки для роз-міщення власної компоненти».	4
Модульний контроль	20
Виконання індивідуального завдання (СР)	30
Разом: Аудиторна робота	**70**
Самостійна робота (СР)	**30**
Всього:	**100**

КОНТРОЛЬ ТА КРИТЕРІЇ ОЦІНЮВАННЯ ЗНАНЬ СТУДЕНТІВ

При вивченні дисципліни використовуються наступні форми контролю знань студентів: поточний; модульний; підсумковий.

Поточний контроль передбачає перевірку теоретичних питань, самостійної роботи, практичних робіт та усне опитування по кожній практичній роботі. По даному виду контролю оцінювання знань здійснюється у відповідності до бального розподілу наведеного в попередній таблиці.

Модульний контроль передбачає виконання модульної контрольної роботи. Всі завдання оцінюються в 20 балів. Перше завдання (теоретичне) – 4 бали, друге завдання (практичне) – 8 балів, третє завдання (практичне) – 8 балів.

Формою підсумкового контролю є екзамен. Екзаменаційна оцінка (100 балів) є результатом виконання двох теоретичних питань (2 х 20 балів = 40 балів) та практичного завдання (60 балів).

Результуюча оцінка з дисципліни визначається як середня від балів набраних протягом семестру та отриманих на іспиті.

СПИСОК РЕКОМЕНДОВАНИХ ДЖЕРЕЛ

Основний:

1. Грицюк Ю Програмування мовою С++ / Ю.Грицюк, Т.Рак навчальний посібник. – Львів : Вид-во Львівського ДУ БЖД, 2011. – 292 с.
2. Бєлов Ю. Вступ до програмування мовою С++. Організація обчислень: навч. посіб. / Ю.Бєлов, Т.Карнаух, Ю.Коваль, А.Ставровський. – К. : Видавничо-поліграфічний центр "Київський університет", 2012. – 175 с. с.: іл. (укр.)
3. Stephen D.Gilbert, Bill McCarty Visual C++ Programming Blue Book. — The Coriolis Group., Inc.,2015. — 14455 N Hayden Drive., Suite 200 Arizona 85260 USA.
4. Sedgewick R., Algorithms in C++: Fundamentals, data structurs, sorting, serching / Princeton University 2011.– 1056 с.
5. A Tour of C++ (2nd Edition) (C++ In-Depth Series) / by Bjarne Stroustrup. – Addison-Wesley Professional, July 9, 2018. – 256 p.