ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КАФЕДРА МОДЕЛЮВАННЯ ТА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Силабус навчальної дисципліни «Основи програмування»	
1. Загальна інформація	
Освітня програма (галузь, спеціальність, рівень вищої освіти, форма навчання)	Освітня програма: «Інженерія програмного забезпечення» Галузь знань: 12 «Інформаційні технології» Спеціальність: 121 «Інженерія програмного забезпечення» Рівень вищої освіти: Перший (бакалаврський) Форма навчання: Денна, заочна
Тип дисципліни (нормативна/вибіркова)	Нормативна
Кількість кредитів ECTS та кількість годин денна/заочна (лекції / лабораторні / самостійна робота здобувачів) **Форма контролю**	Кредити – 12,0 Загальний обсяг – 360 год., двосеместрова дисципліна Денна: 68 лекц., 68 лаб. роб., 224 сам. роб. Заочна: 16 лекц., 16 лаб. роб., 328 сам. роб. Форма контролю – залік, екзамен
Викладачі (ПІП, наукові ступені і звання, контактний e-mail)	Рибальченко Олена Геннадіївна, ст. викладач rybalchenko@knu.edu.ua
Посилання на матеріали дисципліни (робоча програма, методичні матеріали)	https://drive.google.com/drive/folders/1WRNDyTTxtcaJM5Hkl4Ckmc-UCyuTc05f?usp=sharing
Мова викладання	Українська
Інформація про розклад занять	http://asu.knu.edu.ua/time-table/chair
Кафедра: (адреса, телефон, e-mail, сайт, QR-code)	вул. Віталія Матусевича, 11, корпус 1, каб. 317 м. Кривий Ріг; тел. 056-409-06-07 mpz@knu.edu.ua http://mpz.knu.edu.ua/
2. Коротка анотація до курсу	
Вивчення дисципліни «Основи програмування» охоплює отримання теоретичних знань та практичних навичок з алгоритмізації та програмування мовами високого рівня C++ та Python, навчання створенню програмних проектів в IDE Visual Studio та PyCharm. Отриманий досвід використовується для вибору оптимального алгоритму вирішення задачі та оптимальної мови програмування, виявлення слабких місць реалізації функціонування системи. Здобуті знання та навички у галузі програмування є базою для вивчення дисциплін професійно-орієнтованого циклу.	
3. Мета та завдання курсу	
Метою дисципліни є вивчення принципів обробки інформації на комп'ютері, формування у студентів навичок розробки та відлагодження прикладних програм мовами програмування C++ та Python із застосуванням технологій структурного, модульного та об'єктно-орієнтованого програмування, отримання практичних навичок використання інтегрованих середовищ розробки програмного забезпечення. Основними завданнями вивчення дисципліни є оволодіння навичками формалізації задач; оволодіння базовими методами проектування алгоритмів і програм; вивчення основних методологій розробки програмного забезпечення; отримання навичок розробки програм мовами програмування C++ та Python; вивчення основ управління проектами розробки програмного забезпечення; вивчення основ забезпечення та контролю якості програмного забезпечення.	

4. Що ви будете знати	5. Що ви будете вміти
• основні синтаксичні конструкції мов програмування C++ та Python; • основні функції стандартних бібліотек мов програмування C++ та Python; • основні методології розробки програмного забезпечення; • основи управління проектами розробки програмного забезпечення; • основи забезпечення та контролю якості програмного забезпечення.	• розробляти структуровані алгоритми розв'язання задач, використовуючи типові алгоритмічні конструкції; • створювати програми мовами програмування C++ та Python із застосуванням технологій структурного, модульного та об'єктно-орієнтованого програмування; • відлагоджувати та тестувати програми для одержання коректних результатів; • користуватися інтегрованими середовищами розробки програмного забезпечення.

6. Матеріально-технічне / інформаційне та навчально-методичне забезпечення

1. Лекційна аудиторія з мультимедійним проектором та підключенням до мережі Інтернет.
2. Аудиторія персональних комп'ютерів з операційною системою типу Windows 7, 8 або 10 та підключенням до мережі Інтернет.
3. Програмне забезпечення: Visual Studio від 2015, Python 3, IDE PyCharm.
4. Основи алгоритмізації та програмування мовами C++, Visual Basic, Turbo Pascal: навч. посіб. / Азарян А. А., Карабут Н. О., Козикова Т. П., Рибальченко О. Г., Трачук А. А., Шаповалова Н. Н. Кривий Ріг, 2014. 308 с.
5. Базові алгоритми та основи програмування. Теорія і практика: навч. посіб. / О.Є. Тверитникова, В. А. Крилова, О. Г. Васильченков. Нац. техн. ун-т "Харків. політехн. ін-т". Харків : Панов А. М., 2020. 264 с.
6. Зеленяк О. Програмування мовою Python. Алгоритмічні структури і стратегії. Теорія та практика. Київ: Ліра-К, 2024. 338 с.
7. Schildt H. C++ The Complete Reference. 5th Edition. McGraw-Hill Osborne Media. 2014. 805 p.
8. Rao S. C++ in One Hour a Day, Sams Teach Yourself. 9th Edition. Sams Publishing, 2022. 848 p.
9. Марк Лутц. Вивчаємо Python. 5-е видання. Київ: Діалектика, 2019, 832 с.
10. Рибальченко О.Г., Білашенко С.В. Методичні вказівки до виконання лабораторних робіт з дисципліни «Основи програмування, 1 частина» для студ. всіх форм навч. за спец. 121 «Інженерія програмного забезпечення». Кривий Ріг, 2023. 72 с.
11. Рибальченко О.Г., Білашенко С.В. Методичні вказівки до виконання лабораторних робіт з дисципліни «Основи програмування, 2 частина» для студ. всіх форм навч. за спец. 121 «Інженерія програмного забезпечення». Кривий Ріг, 2024. 48 с.
12. A Byte of Python. URL: https://python.swaroopch.com/ (дата звернення 20.12.2024).
13. Dive Into Python. URL: https://diveintopython.org/ (дата звернення 20.12.2024).
14. Клас з C++ https://classroom.google.com/c/MTYzMzMxMDIwNTMy?hl=ru&cjc=5j2rtbd.
15. Клас з Python https://classroom.google.com/c/NTMzNDU5NDg2MDFa?hl=ru&cjc=gmadc23.

7. Тематика курсу

Змістовий модуль 1. Основи структурного програмування мовою С++

Історія розвитку мови С++. Структурне, модульне та об'єктно-орієнтоване програмування. Поняття компілятора, інтерпретатора. Структура програми на С++ та етапи її обробки. Препроцесор. Заголовні файли. Директива #include. Використання об'єктів cin та cout. Коментарі та їх використання. Призначення функції main(). Стандартні функції та їх використання.

Змінні. Виділення пам'яті. Розмір змінних і його визначення за допомогою sizeof. Знакові і беззнакові величини (signed /unsigned). Основні типи змінних (int, long, char, float, double, bool). Розмір в пам'яті і допустимий діапазон значень. Оголошення змінних. Чутливість до регістра букв. Одночасне оголошення більш ніж однієї змінної. Надання значення змінним. Ініціалізація. Оператор typedef. Використання типів int, short та long. Переповнювання беззнакових і знакових цілих. Символи. Тип char і числа. Спеціальні символи. Константи. Константи-літерали. Символічні константи. Визначення констант за допомогою #define і const. Константи, що перераховуються. Оператор enum.

Оператори. Порожні символи. Блоки і складені оператори. Вирази. Операція присвоювання. Математичні операції. Цілочисельне ділення і залишок від ділення. Поєднання присвоювання і математичних операцій. Інкремент і декремент. Префіксні і постфіксні операції. Пріоритет виконання операцій. Вкладені дужки. Операції відношення.

Простий оператор if. Стилі виділення відступами. Повна форма оператора if з ключовим словом else. Вкладені оператори if. Використання дужок у вкладених операторах if. Логічні оператори І, АБО, НЕ (&&, ||, !). Пріоритет операцій відношення і логічних операцій. Умовна (тернарна) операція. Оператор switch. Використання ключових слів case, default і break. Використання оператора switch в програмах, що містять меню. Цикли. Цикл з ключовим слово goto. Цикли while. Цикли while із складними умовами. Оператори continue та break. Цикли do-while. Цикли for. Множинна ініціалізація та інкрементування в циклах for. Видимість імен в циклах. Безконечні цикли.

Змістовий модуль 2. Функції та управління пам'яттю у мові С++

Що таке покажчик? Адреси змінних. Використання операції & (адреса). Зберігання адреси в покажчику. Імена покажчиків. Непряма адресація. Операція *. Покажчики, адреси і змінні. Маніпулювання даними за допомогою покажчиків. Стек і вільна пам'ять (heap). Виділення, використання і видалення змінних у вільній пам'яті. Оператори new і delete. Null-покажчик. Покажчики const. Витік пам'яті. Створення і видалення об'єктів у вільній пам'яті. Покажчик this. Створення і використання посилань. Використання оператора адреси до посилань.

Масиви як структурований тип даних. Визначення масивів. Синтаксис об'яви масивів. Поняття "елемент масиву" та "індекс масиву". Синтаксис використання масивів. Зв'язок покажчиків з масивами. Вирази та арифметика з покажчиками. Динамічний розподіл пам'яті. Стандартні алгоритми роботи з масивами: визначення екстремальних значень, статистичних показників, упорядкування масивів, пошук визначеного елементу. Багатовимірні масиви. Приклади програм з багатовимірними масивами. Реалізація алгоритмів, пов'язаних з обробкою матриць.

Програмні модулі мови С++. Функції математичної бібліотеки. Функції бібліотеки стандартного введення/виведення. Використання функцій при розробці програм на мові С++. Визначення функцій. Прототипи функцій та файли заголовків. Розробка власних функцій. Виклик функції за значенням. Способи обміну інформацією між функціями. Локальні та глобальні змінні. Виклик функції за посиланням –передача параметрів за адресою. Приклад програми, що використовує виклик за посиланням. Класи пам'яті. Правила області дій. Параметри за умовчанням (default). Перевантаження (поліморфізм) функцій. Розробка програми у вигляді проекту. Зовнішні змінні. Рекурсія. Приклад використання рекурсії: числа Фібоначчі. Перевантаження функцій. Шаблони функцій.

Масиви символів (рядки С-типа). Використання cin, cin.get() для введення рядків. Використання бібліотеки string.h для роботи з рядками. Визначення довжини рядка за допомогою strlen(). Присвоєння (копіювання) рядків функціями strcpy() і strncpy(). Порівняння рядків за допомогою strcat() і strncat(). Основні функції для рядків С-стилю.

Змістовий модуль 3. Об’єктно-орієнтоване програмування на С++

Створення нових типів даних. Класи і члени. Оголошення класу. Визначення об'єкту. Класи і об'єкти. Доступ до членів класу. Ключові слова private і public. Обґрунтування необхідності в приватних членах-даних. Приватність і безпека. Реалізація методів класу. Конструктори і деструктори. Конструктори і деструктори за умовчанням. Використання покажчика this. Функції-члени const. Інтерфейс і реалізація. Реалізація inline. Класи з іншими класами як дані члени. Структури.

Спадкування. Базові і похідні класи. Синтаксис похідних класів. Private та protected. Конструктори і деструктори в похідних об'єктах. Передача аргументів в базові конструктори. Перевантаження конструкторів в похідних класах. Перекривання методу базового класу в похідному. Утаєння методів базового класу. Віртуальні методи. Раннє та пізнє зв’язування. Віртуальні деструктори. Віртуальний конструктор копіювання. Поліморфізм. Ідентифікація типа під час виконання. Множинне спадкування. Віртуальне спадкування. Абстрактні типи даних. Повністю віртуальні функції. Складні ієрархії абстракцій.

Потоки. Буферизація. Стандартні об'єкти введення/виводу. Введення з використанням cin. Використання cin.get(). Введення рядків із стандартного введення. Використання cin.getline(). Использование cin.ignore(). Використання cin.peek() і cin.putback(). Виведення з використанням cout. Очищення виводу. Метод cout.flush(). Функції cout.put() і cout.write(). Маніпулятори, прапори і інструкції форматування. Використання cout.width(), cout.fill(), cout.setf(). Потоки проти функції printf. Файлове введення/виведення. Об'єкти типа ofstream і ifstream. Відкриття та закриття файлів для введення і виводу. Функції eof(), bad(), fail(), good(). Зміна поведінки ofstream при відкритті файлів. Використання ios::app, ios::ate, ios::trunc, ios::nocreate, ios::noreplace.

Змістовий модуль 4. Програмування мовою Python.

Історія створення мови Python. Інтерпретатор Python. Інтерактивний та сценарний режими. Інтегровані середовища розробки для Python. Введення та виведення даних, форматне виведення. Динамічна типізація в мові Python. Числа, рядки, послідовності, відображення. Арифметичні вирази. Складені об’єкти: списки, кортежі, словники, множини. Робота з файлами.

Структури управління та повторення у мові Python. Повне та неповне розгалуження. Реалізація багатоваріантного вибору через розгалуження. Параметричний цикл з варіативною та без варіативної частини. Цикл з передумовою. Оператори continue та break. Виконання команд всередині контексту (with). Обробка виключень в програмах, вбудовані типи виключень та визначення нових виключень.

Визначення власних функцій. Передача параметрів і повернення результатів. Значення аргументів функції за замовчуванням. Довільний набір аргументів. Іменовані аргументи. Правила видимості. Функції, як об'єкти та замикання. Області видимості у мові Python. Декоратори. Ітератори. Генератори та со-програми. Генератори списків. Вирази-генератори. Основи декларативного програмування. Оператор Lambda. Атрибути функцій. Розробка рекурсивної функції на базі її процедурної форми. Послідовна, паралельна та псевдопаралельна рекурсії. Обробка списків рекурсивними функціями.

Створення і використання модуля. Пошук модулів і компільовані файли. Стандартні модулі: sys, os. Пакети. Коротка характеристика нестандартних модулів Python. Модуль чисел з плаваючою точкою Decimal. Модуль раціональних чисел Fractions. Модуль стандартних математичних функцій Math. Модуль абстрактних базових класів Numbers. Модуль псевдовипадкових чисел Random. Модуль для роботи з комплексними числами CMath. Модуль для роботи з масивами Array. Модуль сортування списків Bisect. Стандартні модулі обробки рядків String і Codecs. Модуль приблизного порівняння двох рядків DiffLib. Модуль для роботи з кодуванням і регулярними виразами Re. Модуль перетворення даних Struct. Модуль доступу до бази символів UnicodeData.

Програмування класів. Методи класів. Екземпляри класів. Спадкування. Простір імен. Перевантаження операторів. Менеджери контексту. Обробка виключень в програмах, вбудовані типи виключень та визначення нових виключень.

8. Система оцінювання

Використовується модульно-рейтингова система оцінювання, яка передбачає розподіл балів за виконання всіх запланованих видів робіт. При цьому максимальна кількість балів за модуль дорівнює 100 Ця сума складається з балів, що їх накопичив студент у ході виконання лабораторних робіт та оцінки за контрольно-модульну роботу.

Лабораторні роботи у модулі відображують оволодіння навичками та вміння застосовувати знання на практиці і сукупно відповідають 75-ти відсоткам ваги. При зниженні якості виконання лабораторної роботи, знижується і кількість балів, якою вона оцінюється. Оцінювання кожної лабораторної роботи ведеться за наступними показниками:

1. Своєчасність практичного виконання лабораторної роботи (згідно із графіком робіт) (0-3 бали).
2. Якість захисту виконаної лабораторної роботи (0-3 бали).
3. Якість знайдених студентом рішень (ефективність алгоритму, доречність використання елементів інтерфейсу, тощо) (1-9 балів).

Якість знайдених студентом рішень оцінюється наступним чином:

- робота виконана без зауважень - максимальний бал;
- робота виконана достатньо повно з деякими зауваженнями –75% від максимального бала;
- робота виконана не повністю – 50% від максимального бала.

Оцінка всієї лабораторної роботи знаходиться підсумовуванням балів за кожний з показників.

Модульний контроль здійснюється 2 рази за семестр. Модульна робота складається з контрольних питань та завдання. Оцінювання питань розподіляється пропорційно їх кількості.

Семестровий контроль здійснюється у формі заліку після першого семестру вивчення дисципліни і екзамену після другого семестру вивчення дисципліни. Для допуску до підсумкового контролю студент повинен виконати графік навчального процесу, усі види запланованих завдань і протягом семестру отримати в сумі не менше 50 балів за кожен зі змістових модулів.

У разі виконання студентом усіх видів поточних контрольних заходів залікова оцінка виставляється студенту на підставі середнього арифметичного зарахованих протягом семестру балів.

Результати екзамену оцінюються за 15-бальною шкалою. Підсумкова оцінка з дисципліни розраховується як середня зважена з оцінок по модулях, включаючи екзаменаційну.

9. Зарахування результатів неформальної освіти

Окремі модулі курсу можуть бути зараховані за умови надання здобувачем вищої освіти сертифікату про проходження он-лайн курсів, зазначених в робочій програмі дисципліни.

10. Політика курсу

Відвідування занять. Регуляція пропусків. Відвідування усіх занять є обов'язковим. У випадку пропуску лабораторного заняття, студент має виконати та здати лабораторну роботу згідно графіку, наведеного у робочій програмі дисципліни. У випадку пропуску лекції студент опрацьовує матеріал самостійно та може задати питання на консультації.

Політика академічної доброчесності регламентується Положенням про академічну доброчесність у Криворізькому національному університеті.

Використання комп'ютерів/телефонів на занятті. Використання комп'ютерів на практичних заняттях є обов'язковим задля досягнення навчальної мети.

Розробник силабусу:
Старший викладач кафедри моделювання та програмного забезпечення
Олена РИБАЛЬЧЕНКО

Завідувач кафедри моделювання та програмного забезпечення, доцент, канд. пед. наук
Андрій СТРЮК

Гарант ОПП, канд. пед. наук
Андрій СТРЮК