

**ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ І.І. МЕЧНИКОВА
ФАКУЛЬТЕТ МАТЕМАТИКИ, ФІЗИКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА МАТЕМАТИЧНОГО ЗАБЕЗПЕЧЕННЯ КОМП'ЮТЕРНИХ СИСТЕМ**

Силабус курсу «Програмування»

| | |
|---------------------------|---|
| Обсяг | загальна кількість: кредитів – 12,0; годин – 360; змістових модулів – 6 |
| Семестр | осінній, весняний |
| Дні, Час, Місце | за розкладом занять |
| Викладач(и) | Антоненко Олександр Сергійович, кандидат фізико-математичних наук, доцент кафедри математичного забезпечення комп'ютерних систем |
| Контактний телефон | (048)7340723 |
| E-mail | asantonenko@gmail.com ; antonenko@onu.edu.ua |
| Робоче місце | кафедра математичного забезпечення комп'ютерних систем |
| Консультації | очні консультації: розклад консультацій видається на початку занять on-line консультації: ZOOM (посилання генерується на початку занять) |

КОМУНІКАЦІЯ

Комунікація зі студентами буде здійснюватися електронною поштою, за допомогою месенджера Telegram, в аудиторії або через ZOOM.

АНОТАЦІЯ КУРСУ

Предметом вивчення курсу є програмні систем, розроблені в структурно-процедурному та об'єктно-орієнтованому стилі, а також їх складові – типи даних, змінні, константи, функції, класи, об'єкти, поля, методи, тощо.

Пререквізити курсу

Матеріал курсу *ґрунтується* на раніше отриманих студентами знаннях, практичних вміннях та навичках з тем та напрямів щодо сучасних операційних систем, а також засвоєних з шкільного курсу математики та інформатики.

Постреквізити курсу

Цей курс є основою для засвоєння наступних дисциплін освітньо-професійної програми підготовки бакалаврів за спеціальністю 123 «Комп'ютерна інженерія»: «Системне програмне забезпечення», «Системне програмування», «Структури даних та алгоритми», «Інженерія програмного забезпечення», «Комп'ютерні системи штучного інтелекту», «Організація баз даних», «Технологія проектування комп'ютерних систем», «Архітектура комп'ютерів та низькорівневе програмування», «Захист інформації у комп'ютерних системах».

Метою курсу є придбання базових знань і практичних навичок розробки та налагодження програм на мовах програмування високого рівня відповідно до сучасних технологій та парадигм структурно-процедурного та об'єктно-орієнтованого програмування, вивчення теоретичних основ алгоритмізації і проектування програм, оволодіння базовими алгоритмами, навичками алгоритмічного мислення, навичками розв'язування простіших задач науково-технічного характеру та змісту, засвоєння методів тестування програм та оформлення відповідних презентацій та звітів про виконану роботу, формування у студентів систематичного, інженерного, самостійного, наукового та творчого підходу до освоєння технологій, методів і засобів виробництва програмного забезпечення.

Завдання:

- освоєння та використання на практиці сучасних технологій програмування (структурне, модульне, об'єктно-орієнтоване програмування);
- освоєння принципів проектування алгоритмів;
- вивчення теоретичних основ алгоритмізації і проектування програм;

- вивчення та практичне засвоєння студентами об'єктно-орієнтованого способу розробки програмних систем;
- навчання студентів використанню сучасного інструментарія розробки, у тому числі: мов програмування C++ та C#, бібліотеки класів STL, інтегрованого середовища програмування Visual Studio чи аналогічного.

Зміст курсу

Розглядаються: введення в архітектуру комп'ютера, системи числення, машинні мови та мови високого рівня, інтерпретатори і компілятори, парадигми програмування, життєвий цикл програми, моделі програмних систем, алгоритми, їх способи запису, алфавіт та лексеми мов C/C++, ідентифікатори, ключові слова, літерали, змінні та типи, вбудовані типи мов C/C++ та їх літерали, операції та вирази, оператори мов C/C++, поняття о структурному програмуванні, оператори розгалуження, оператори циклу, оператори переходу, масиви, контейнери мови C++, алгоритми пошуку і сортування, функції, параметри функції, їх передача у функції, рядковий тип, двовимірні та багатовимірні масиви, вказівники, посилання та керування пам'яттю, перевантаження функцій, рекурсивні функції, шаблонні функції, основні поняття об'єктно-орієнтованого програмування, об'єктна декомпозиція, поняття класу і об'єкта, поля й методи класів, конструктори та деструктори, інкапсуляція, успадкування, поліморфізм, абстрактні класи, інтерфейси, діаграми класів UML, організація C++-системи введення-виведення, простори імен, виняткові ситуації, абстрактні типи даних (АТД), реалізація АТД у вигляді класів, шаблонні функції та класи, стандартна бібліотека шаблонів (STL).

ОЧІКУВАНІ РЕЗУЛЬТАТИ

У результаті вивчення курсу студент повинен

знати: поняття інформації та способи її обробки та використання; системи числення, принципи представлення у комп'ютері цілих і дійсних чисел, а також символів і рядків; мови низького і високого рівнів; поняття інтерпретації і компіляції програм; основні етапи життєвого циклу програми: постановка задачі, проектування, кодування, тестування та налагодження, експлуатація; основні способи запису алгоритмів; базові алгоритмічні конструкції: послідовне виконання, розгалуження, цикл; основні операції та оператори мов C/C++; основні скалярні типи даних мов C/C++; структуровані типи даних: контейнерні типи, масиви, рядки, структури, класи, списки, файли; поняття функції, механізми передачі параметрів у функції; прийоми створення рекурсивних структур і функцій; команди передпроцесорної обробки мови C/C++; модульний принцип розробки програм; способи виділення пам'яті; правила визначення класів в мові C++ та C#, способи взаємодії об'єктів в програмній системі; основні принципи об'єктно-орієнтованого підходу до програмування – абстракція, інкапсуляція, успадкування і поліморфізм; правила побудови АТД; як будувати класи, які повністю або частково моделюють АТД.

вміти: розробляти алгоритми вирішення обчислювальних задач; складати лінійні програми мовами C/C++; складати циклічні програми мовами C/C++; працювати із скалярними даними: цілими, дійсними, символьними, логічними; складати програми обробки масивів даних; використовувати функції мов C/C++ при складанні програм; використовувати структуровані типи даних: масиви, рядки, структури, класи, списки, файли; використовувати стандартні функції мов C/C++; виконувати введення-виведення даних; реалізовувати багатомодульні програми; складати алгоритми пошуку та сортування масивів; користуватися засобами інтегрованого середовища розробки Microsoft Visual Studio, Code::Blocks, onlineGDB чи аналогічного; здійснювати функціональну та об'єктну декомпозицію програми відповідно до обраної технології програмування; застосовувати прийоми об'єктно-орієнтованого програмування на практиці; кодувати програми на об'єктно-орієнтованих мовах програмування, таких як C++ та C#; використовувати довідкову систему, офіційну документацію та інші онлайн джерела для отримання докладних відомостей про використовувані класи; виправляти синтаксичні та семантичні помилки та здійснювати рефакторинг коду (налагоджувати та тестувати програму); використовувати при написанні програм основні стандартні класи STL; використовувати при написанні програм засоби обробки виняткових ситуацій і інші засоби підвищення якості і надійності програмного коду; створювати програмні системи відповідно до принципів об'єктно-орієнтованого програмування.

Компетентності, які отримує студент у результаті вивчення курсу:

- вміння виявляти, ставити та вирішувати проблеми;
- здатність використовувати сучасні методи і мови програмування для розроблення алгоритмічного та програмного забезпечення;
- здатність створювати системне та прикладне програмне забезпечення комп'ютерних систем та мереж;
- здатність оформляти отримані робочі результати у вигляді презентацій, науково-технічних звітів.

Результати навчання: по завершенню курсу студент матиме навички

- знати і розуміти наукові положення, що лежать в основі функціонування комп'ютерних засобів, систем та мереж;
- вміти застосовувати знання для ідентифікації, формулювання і розв'язування технічних задач спеціальності, використовуючи методи, що є найбільш придатними для досягнення поставлених цілей;
- вміти системно мислити та застосовувати творчі здібності до формування нових ідей;
- вміти здійснювати пошук інформації в різних джерелах для розв'язання задач комп'ютерної інженерії;
- вміти ефективно працювати як індивідуально, так і у складі команди;
- вміти оцінювати отримані результати та аргументовано захищати прийняті рішення.

ФОРМИ І МЕТОДИ НАВЧАННЯ

Курс буде викладений у формі лекцій (96 год.) та лабораторних занять (80 год.), організації самостійної роботи студентів (184 год.).

Основна підготовка студентів здійснюється на лекційних та лабораторних заняттях.

Під час викладання курсу використовуються такі **методи навчання**: *словесні* (лекція, пояснення); *наочні* (презентація); *практичні* (лабораторні роботи); *робота з літературними джерелами* (самостійна робота студентів), *виконання індивідуального навчально-дослідного завдання* (самостійна робота студентів).