

СИЛАБУС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ ОБЧИСЛЮВАЛЬНА ТЕХНІКА ТА ПРОГРАМУВАННЯ





Ступінь освіти	бакалавр
Освітня програма	Електроенергетика, електротехніка та електромеханіка
Тривалість викладання	1, 2, 3 чверть
Заняття:	Осінній та весняний семестр
Лекції (год/тижд.):	2 год. (1 чв.), 2 год. (2 чв.), 1 год. (3 чв.)
Лабораторні заняття (год/тижд.):	2 год. (1 чв.), 2 год. (2 чв.), 1 год. (3 чв.)
Практичні заняття (год/тижд.):	1 год. (1 чв.), 1 год. (2 чв.), 1 год. (3 чв.)
Підсумковий контроль	іспит
Мова викладання	українська

Сторінка курсу в СДО НТУ «ДП»: <https://do.nmu.org.ua/course/view.php?id=3659>

Кафедра, що викладає Інформаційних технологій та комп'ютерної інженерії

Інформація про викладачів:

	Викладачі: Коротенко Григорій Михайлович , доцент, д.т.н., професор кафедри Персональна сторінка http://it.nmu.org.ua/ua/HR_staff/prepods/korotenko.php E-mail: korotenko.g.m@nmu.one
	Зарічний Володимир Сергійович Асистент кафедри E-mail: zarichnyi.v.s@nmu.one

1. Анотація до курсу

Обчислювальна техніка та програмування – комплексна наукова й інженерно-технічна дисципліна, що направлена на формування низки уявлень та сервісно-орієнтованих підходів до опрацювання цифрових даних в умовах глобального комп'ютерингу і необхідності виконання завдань цифрових трансформацій в усіх сферах економіки України (Цифрова агенда України). Це потрібно особливо в задачах розрахунку, розробки та використання

різномірних компонентів електронної техніки (або електротехніки), проектування для створених зразків програмного забезпечення та подальшої їхньої програмно-апаратної інтеграції бакалаврами спеціальності «Електроенергетика, електротехніка та електромеханіка» з метою формування інноваційних компетентностей майбутніх фахівців у роботі з цифровими даними.

2. Мета та завдання курсу

Мета дисципліни – теоретична і практична підготовка студентів щодо використання моделей і методів комп'ютерного для створення, зберігання, обробки та аналізу цифрових даних на комплексі сучасного апаратного і програмного забезпечення, який здатний організувати доступ до сучасних інформаційних ресурсів, а також надавати ефективні засоби і методи формування відповідного цифрового оточення.

Завдання курсу:

- ознайомити слухачів з основними характеристиками технічних пристроїв комп'ютерних засобів;
- підвищити ефективність застосування студентами прикладного програмного забезпечення для обробки цифрової інформації;
- долучити здобувачів вищої освіти до основ комп'ютерного і технологій використання пакетів прикладних програм для обробки текстової, табличної, графічної та математичної інформації при вирішенні фахових задач;
- засвоїти основні поняття програмування, базові типи даних, операції й засоби організації обчислень мовою C++, а також елементи технології створення програм.

3. Результати навчання

Демонструвати навички використання цифрових систем та технологій, а також володіти елементами рішення різних задач за допомогою мови програмування C++ для подальшого використання у професійній діяльності та у соціальному контексті.

4. Структура курсу

ЛЕКЦІЇ

1. Головні компоненти обчислювальної техніки та їх взаємодія в процесах повсюдного комп'ютерного.

1.1 Поняття інтерфейсу. Рівні та види інтерфейсів апаратних та програмних складових.

1.2 Архітектура структур і складових обчислювальної техніки.

1.3 Фреймворки архітектур як еталонні моделі організації різних елементів обчислювальної техніки.

1.4 Платформорієнтований розвиток обчислювальної техніки: технологічна база для розробки та взаємодії програмних додатків.

2. Програмне забезпечення для апаратно-програмної інтеграції функціонуючих елементів обчислювальної техніки.

2.1 Основні поняття, склад та структура програмного забезпечення.

2.2 Прикладне програмне забезпечення: класифікація за типом та сферою застосування, а також способом запуску.

2.3. Функціонування програм, додатків, застосунків, програмних одиниць та програмних продуктів в структурі компонентів обчислювальної техніки.

3. Прикладне програмне забезпечення загального призначення.

3.1 Основні компоненти інтегрованого пакету MS Office.

4. Основи алгоритмізації.

4.1 Поняття алгоритму: його властивості та способи зображення. Блок-схеми алгоритмів.

4.2 Особливості комп'ютера як виконавця алгоритмів.

4.3 Метод покрокової деталізації при розробці алгоритмів.

4.4 Базові структури опису алгоритмів та особливості їх використання.

5. Основні поняття процесу розробки програмного забезпечення для ЕОМ.

5.1 Особливості застосування технологій програмування.

5.2 Процедурне та об'єктно-орієнтоване програмування.

5.3 Види і склад систем програмування.

5.4 Машинні мови, мови асемблера та мови високого рівня.

5.5 Загальне та різне у мовах C і C++.

5.6 Основи конструювання програм на мові C++.

6. Парадигми мов програмування.

6.1 Парадигми програмування, які підтримує мова C++.

6.2 Середовища для розробки програм мовою C++.

6.3 Приклади програм: друкування рядка тексту та складання чисел.

6.4 Організація пам'яті. Формування арифметичних виразів з чисел, змінних та функцій, які з'єднані між собою знаками арифметичних операцій.

6.5 Пріоритети різних за призначенням операцій у виразах.

7. Введення в мову програмування C++.

7.1 Загальна характеристика мови.

7.2 Використання функції *main()*.

7.3 Заголовок функції як засіб інтерфейсу.

7.4 Коментарі у мові C++.

7.5 Препроцесор C++ і файл *iostream*.

8. Процеси вводу та виводу даних у програмах на мові C++.

8.1 Імена заголовкових файлів.

8.2 Області імен.

8.3 Символ нової строки (**\n**).

8.4 Вивод даних в мові C++ з використанням об'єкта *cout*.

8.5 Форматування вихідного коду C++.

9. Алфавіт, ідентифікатори, лексеми і токени мови C++.

9.1 Оператори оголошення та змінні.

9.2 Оператор привласнення: різновиди та специфіка використання.

9.3 Операції, операнди, вирази й оператори.

9.4 Поняття пріоритетів виконання операцій у складі операторів.

10. Типи даних і використання змінних у мові C++.

10.1 Класифікація типів даних.

10.2 Оголошення змінних.

10.3 Визначення типів змінних.

10.4 Завдання констант.

10.5 Час існування і область видимості змінних.

11. Функція як один з важливіших компонентів мови C++.

11.1 Типи функцій з точки зору визначення типів значень, які вони повертають.

11.2 Застосування функцій, які повертають значення.

11.3 Різновиди використовуваних функцій.

11.4 Функції, які визначаються користувачем.

11.5 Формат визначення функції.

11.6 Заголовки функцій.

11.7 Функція, що визначається користувачем та повертає значення.

12. Введення у використання управляючих операторів.

12.1 Алгоритми реалізації управляючих операторів.

12.2 Псевдокоди та блок-схеми.

12.3 Види управляючих структур.

12.4 Оператор вибору *if*.

12.5 Типові приклади застосування управляючих операторів.

13. Складні конструкції управляючих операторів.

13.1 Оператор вибору *if...else*.

13.2 Оператор вибору *while*.

13.3 Типові приклади застосування складних управляючих операторів.

13.4 Особливості алгоритмічного уявлення складних операторів.

13.5 Приклади використання.

14. Формулювання циклічних алгоритмів: повторення, що управляються лічильником.

14.1 Формування алгоритмів повторення, що управляються оператором.

14.2 Оператори привласнення.

14.3 Множина операцій мови C++: операції інкремента та декремента, а також інфіксні, префіксні, постфіксні та тернарні (умовні триарні або тримісні) операції.

14.4 Приклади використання циклічних операторів для вирішення типових завдань.

15. Різновиди управляючих операторів повторення.

15.1 Основи створення конструкцій з повторенням, що управляються лічильником.

15.2 Оператор повторення *for*.

15.3 Приклади використання операторів (циклів) *for*.

15.4 Оператор повторення (циклів) *do...while*.

15.5 Оператори *break* и *continue*.

15.6 Вкладені цикли.

15.7 Рекомендації щодо вибору та застосування циклів.

15.8 Керуючі оператори в циклах.

15.9 Приклади використання управляючих операторів повторення.

16. Компоненти програм на мові C++.

16.1 Функції математичної бібліотеки.

16.2 Головні компоненти програм при використанні процедурної парадигми у мові C++.

16.3 Визначення функцій з декількома параметрами.

16.4 Прототипи функцій і примусове приведення аргументів.

16.5 Заголовкові файли стандартної бібліотеки C++.

16.6 Приклад генерації випадкових чисел.

16.7 Функції з пустим списком параметрів.

16.8 Перевантаження функцій.

17. Використання масивів і векторів.

17.1 Оголошення та ініціалізація масивів різної вимірності.

17.2 Приклади використання масивів для вирішення різних завдань.

17.3 Одновимірні та двовимірні масиви. Приклади використання.

17.4 Типові алгоритми обробки одновимірних та двовимірних масивів.

17.5 Пошук елементів в масивах. Сортування масивів. Особливості опрацювання багатовимірних масивів.

ЛАБОРАТОРНІ ЗАНЯТТЯ

I-1 – Приклади застосування прикладного програмного забезпечення. Виконання робіт у інтегрованому пакеті MS Office.

I-2 – Знайомлення з середовищем розробки Code::Blocks. Типи проектів. Створення проекту в Code::Blocks. Розробка програми з лінійним обчислювальним процесом.

I-3 – Конструювання програм у середовищі програмування C++ Code::Blocks. Ознайомлення з середовищем розробки Code::Blocks. Типи проектів. Створення проекту в Code::Blocks. Розробка програм з лінійними обчислювальними процесами.

I-4 – Конструювання програм у середовищі програмування C++ Code::Blocks. Розробка програм з розгалужуючими процесами.

I-5 – Конструювання програм у середовищі програмування C++ Code::Blocks. Розробка програми зі складними розгалужуючими процесами.

I-6 – Конструювання програм у середовищі програмування C++ Code::Blocks. Розробка програм з циклічними процесами.

I-7 – Конструювання програм у середовищі програмування C++ Code::Blocks. Розробка програм з циклічними процесами, що включають умови.

I-8 – Конструювання програм у середовищі програмування C++ Code::Blocks. Розробка програм для вирішення практичних задач: табуляція функцій та розрахунок визначених інтегралів.

I-9 – Конструювання програм у середовищі програмування C++ Code::Blocks. Розробка програм з використанням одновимірних масивів.

I-10 – Конструювання програм у середовищі програмування C++ Code::Blocks. Розробка програм з використанням двовимірних масивів.

I-11 – Конструювання програм у середовищі програмування C++ Code::Blocks. Пошук введених користувачем чисел у масивах за певним алгоритмом, сортування даних, що містяться у масивах.

I-12 – Конструювання програм у середовищі програмування C++ Code::Blocks. Розробка програм з двовимірними динамічними масивами.

I-13 – Конструювання програм у середовищі програмування C++ Code::Blocks. Робота з текстовими файлами.

I-14 – Конструювання програм у середовищі програмування C++ Code::Blocks. Робота з бінарними файлами.

5. Технічне обладнання та/або програмне забезпечення

Під час виконання лабораторних робіт використовується пакет програмних застосунків MS Office та вільне кроссплатформенне середовище розробки програм на мові C++ Code::Blocks.

6. Система оцінювання та вимоги

6.1. Навчальні досягнення здобувачів вищої освіти за результатами вивчення курсу оцінюватимуться за шкалою, що наведена нижче:

Рейтингова шкала	Інституційна шкала
90 – 100	Відмінно/ Excellent
75-89	Добре/ Good
60-74	Задовільно/ Satisfactory
0-59	Незадовільно/ Fail

6.2. Лабораторні роботи приймаються за контрольними запитаннями до кожної з роботи. Кожна із 6 робіт оцінюється по 10-бальній шкалі (тобто максимальна сума балів за лабораторні роботи складає 60 балів).

6.3. Теоретична частина оцінюється за результатами здачі наприкінці кожної чверті двох контрольних тестових завдань, кожне з яких містить 25 запитань. За

правильну відповідь на одне запитання студент отримує 2 бали (тобто максимальна оцінка за теоретичну частину – 100 балів).

6.4. Підсумкова оцінка за курс (за 100-бальною шкалою):

$$ПО = \frac{\frac{100}{60} СБ_{лр} \cdot 4 + СБ_{т} \cdot 5}{4 + 5},$$

де $СБ_{лр}$ – сума балів за здачу лабораторних робіт; $СБ_{т}$ – сума балів за теоретичну частину; 60 – максимальна сума балів за лабораторні роботи; 100 – максимальна кількість балів за теоретичну частину; 4 – кількість годин на тиждень лабораторних робіт; 5 – кількість годин на тиждень лекцій.

6.5. Здобувачі вищої освіти можуть отримати підсумкову оцінку з дисципліни на підставі поточного оцінювання знань за умови, якщо набрана кількість балів з поточного тестування з теоретичної частини та лабораторних робіт складатиме не менше 60 балів.

7. Політика курсу

7.1. Політика щодо академічної доброчесності

Академічна доброчесність здобувачів вищої освіти є важливою умовою для опанування результатами навчання за дисципліною і отримання задовільної оцінки з поточного та підсумкового контролів. Академічна доброчесність базується на засудженні практик списування (виконання письмових робіт із залученням зовнішніх джерел інформації, крім дозволених для використання), плагіату (відтворення опублікованих текстів інших авторів без зазначення авторства), фабрикації (вигадування даних чи фактів, що використовуються в освітньому процесі). Політика щодо академічної доброчесності регламентується положенням "Положення про систему запобігання та виявлення плагіату у Національному технічному університеті "Дніпровська політехніка".

http://www.nmu.org.ua/ua/content/activity/us_documents/System_of_prevention_and_detection_of_plagiarism.pdf.

У разі порушення здобувачем вищої освіти академічної доброчесності (списування, плагіат, фабрикація), робота оцінюється незадовільно та має бути виконана повторно. При цьому викладач залишає за собою право змінити тему завдання.

7.2. Комунікаційна політика

Здобувачі вищої освіти повинні мати активовану університетську пошту.

Усі письмові запитання до викладачів стосовно курсу мають надсилатися на університетську електронну пошту.

7.3. Політика щодо перескладання

Роботи, які здаються із порушенням термінів без поважних причин оцінюються на нижчу оцінку. Перескладання відбувається із дозволу деканату за наявності поважних причин (наприклад, лікарняний).

7.4 Політика щодо оскарження оцінювання

Якщо здобувач вищої освіти не згоден з оцінюванням його знань він може опротестувати виставлену викладачем оцінку у встановленому порядку.

7.5. Відвідування занять

Для здобувачів вищої освіти денної форми відвідування занять є обов'язковим. Поважними причинами для неявки на заняття є хвороба, участь в університетських заходах, академічна мобільність, які необхідно підтверджувати документами. Про відсутність на занятті та причини відсутності здобувач вищої освіти має повідомити викладача або особисто, або через старосту.

За об'єктивних причин (наприклад, міжнародна мобільність) навчання може відбуватись в он-лайн формі за погодженням з керівником курсу.

8 Рекомендовані джерела інформації

Базові

1. Microsoft Office 2019 Step by Step / Joan Lambert, Curtis Frye. 1st Edition. Microsoft Press: 2018. 560 p.
2. Stephen Prata. C++ Primer Plus / Sixth Edition. Addison-Wesley Professional: 2011. 1440 p.
3. Harvey M. Deitel, Paul J. Deitel. C++ How no Program / 10th Edition. Pearson Education: 2017. 1074 p.
4. Трофименко О.Г. С++. Алгоритмізація та програмування : підручник / О.Г. Трофименко, Ю.В. Прокоп, Н.І. Логінова, О.В. Задерейко. 2-ге вид. перероб. і доповн. Одеса : Фенікс, 2019. 477 с.
5. Дудзяний І.М.. Програмування мовою С++. Частина 1 : Парадигма процедурного програмування : навчальний посібник. Львів: ЛНУ імені Івана Франка, 2013. 468 с.

Додаткові

1. Algorithm Design. Foundations, Analysis, and Internet Examples / Michael T. Goodrich and Roberto Tamassia. – N.Y.: John Wiley & Sons, Inc., 2014. – 816 p.

9 ІНФОРМАЦІЙНІ РЕСУРСИ

Репозиторій Національного технічного університету «Дніпровська політехніка»: <http://ir.nmu.org.ua/>