



Навчальна дисципліна:

Алгоритмізація та програмування

Вид навчально-методичного забезпечення:

Силабус курсу

для здобувачів першого (бакалаврського) рівня вищої освіти освітньо-професійних програм:

- «Професійна освіта (Комп'ютерні технології)»,
- «Комп'ютерні науки».

Галузь знань:

- 01 Освіта/Педагогіка,
- 12 Інформаційні технології.

Спеціальності:

- 015 Професійна освіта (Комп'ютерні технології),
- 122 Комп'ютерні науки.

Дні занять та консультацій: за поточним розкладом

Рік навчання: I, Семестр: I

Кількість кредитів: 7

Мова викладання: українська

Керівник курсу

д.пед.н., професор кафедри комп'ютерних наук
Луцького національного технічного університету

Тулашвілі Юрій Йосипович

Контактна інформація: y.tulashvili@lutsk-ntu.com.ua

Опис дисципліни

Як навчальна дисципліна, «Алгоритмізація та програмування» забезпечує формування у фахівців комплексу професійних знань щодо засвоєння сучасних методів програмування, а також навичок застосування мов програмування Assembler та C++. Оволодіння методами розробки та використання алгоритмів, структурного та об'єктно-орієнтованого програмування є невід'ємним елементом підготовки висококваліфікованих ІТ фахівців для різних напрямків індустрії програмної продукції в Україні.

По завершенню вивчення дисципліни «Алгоритмізація та програмування» студенти зможуть:

- застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань;
- мотивовано обирати мови програмування та технології розробки для розв'язання завдань створення і супроводження програмного забезпечення;
- володіти мовами системного програмування та методами розробки програм, що взаємодіють з компонентами комп'ютерних систем, знати мережні технології, архітектури комп'ютерних мереж, мати практичні навички технології адміністрування комп'ютерних мереж та їх програмного забезпечення;
- розробляти програмні моделі предметних середовищ, вибирати парадигму програмування з позицій зручності та якості застосування для реалізації методів і алгоритмів розв'язування задач у галузі комп'ютерних наук, створювати надійне та ефективне програмне забезпечення;
- проектувати, розробляти та аналізувати алгоритми розв'язання обчислювальних та логічних задач, оцінювати ефективність та складність алгоритмів на основі застосування формальних моделей алгоритмів та обчислюваних функцій;
- використовувати формальні моделі алгоритмів та обчислюваних функцій, встановлювати розв'язність, часткову розв'язність і нерозв'язність алгоритмічних проблем, проектувати, розробляти й аналізувати алгоритми, оцінювання їх ефективності та складності;
- володіти навичками роботи з сучасними комп'ютерними програмами фахової спрямованості та можливості впровадження відповідного програмного забезпечення для проектування інформаційних систем та педагогічних програмних засобів.

Навчальний матеріал дисципліни структурований за модульним принципом і складається з двох навчальних модулів, а саме:

- навчального модуля № 1 «Введення в мови програмування Assembler та C++».
- навчального модуля № 2 «Основи теорії алгоритмів».

Кожен з модулів є логічно завершеною, відносно самостійною, цілісною частиною навчального плану, засвоєння якого передбачає проведення модульної контрольної роботи та аналіз результатів її виконання.

Структура курсу

Години (лек. / сем.)	Тема	Результати навчання	Завдання
1	2	3	4
2 / 4	1. Принципи машини фон Неймана. Представлення інформації у цифровому вигляді	Розуміти принципи машини фон Неймана, принципи представлення інформації у цифровому вигляді. Знати двійкову та шістнадцяткову системи числення. Вміти оперувати тетрадами, розрізняти додатні та від'ємні двійкові числа, вміти застосовувати прийоми двійкової арифметики. Розуміти принципи кодування символів та вміти визначати їх ASCII коди.	Тести, задачі, питання
2 / 4	2. Мови програмування. Алгоритмічні мови. Машинно-орієнтовні мови програмування. Мови високого рівня.	Знати основи побудови мов програмування. Розрізняти алгоритмічні мови та машинно-орієнтовні мови програмування. Розуміти поняття операнд, оператор, машинна інструкція. Знати як реалізуються трансляція, інтерпретація та компіляція програм.	Тести, задачі, питання
2 / 4	3. Машинно-орієнтовна мова низькорівневого програмування - Assembler.	Розуміти принципи функціонування архітектури центрального процесора, загальні підходи до модифікації мікропроцесора. Знати основи програмування на мові Assembler, формати машинних команд. Вміти виконувати резервування обсягів пам'яті. Розуміти адресацію пам'яті. Знати структуру програми на мові Assembler.	Тести, задачі, питання
2 / 4	4. мова програмування C/C++.	Знати основи програмування на мові C/C++. Розуміти синтаксис, принципи утворення типів даних. Знати форми запису синтаксичних правил, форму Бекуса-Наура, описи синтаксичними діаграмами. Розуміти структуру програми на C++. Вміти програмувати математичні операції.	Тести, задачі, питання
2 / 4	5. Типи даних на мові C++. Простір пам'яті.	Знати типи даних на мові C. Вміти програмувати явне та неявне перетворення типів. Розуміти простір пам'яті. Знати представлення покажчиків у мові C.	Тести, задачі, питання
2 / 4	6. Операції адресної арифметики.	Знати операції адресної арифметики. Вміти програмувати операції адресної арифметики на мові C++. Розуміти адресацію в пам'яті комп'ютера на мові низькорівневого програмування Assembler. Вміти програмувати адресацію по базі на мові Assembler.	Тести, задачі, питання
2 / 4	7. Поняття алгоритму.	Знати поняття алгоритму та властивостей алгоритму. Розуміти класи та види алгоритмів. Знати графічні зображення алгоритмічних блоків. Вміти будувати блок-схеми. Розуміти принципи структурного програмування.	Тести, задачі, питання
2 / 4	8. Типи алгоритмів. Програма із структурою послідовного алгоритму	Знати поняття алгоритму - послідовна структура. Вміти програмувати програми із структурою послідовного алгоритму на мові Assembler та на мові C.	Тести, задачі, питання

2 / 4	9. Програми із структурою розгалуження.	Знати поняття алгоритму - розгалуження. Уміти застосовувати команди передачі управління на мові Assembler. Знати команди умовного переходу. Знати команди з логічними виразами на мові C++, команди розгалуження if (повна, коротка), команду вибору switch, команду безумовного переходу goto. Вміти застосовувати команди умовного переходу у програмах із структурою розгалуження.	Тести, задачі, питання
2 / 4	10. Програми із структурою цикл на мові С.	Знати принципи програмування програм із структурою цикл на мовах Assembler та C++. Вміти застосовувати операції інкременту (++) та декременту (--).Знати команди операторів циклу з лічильником (for), управління лічильником, команду циклу з передумовою (while, команду циклу з післяумовою (do while). Вміти опрацьовувати масиви даних.	Тести, задачі, питання
2 / 4	11. Модульний стиль програмування.	Розуміти модульний стиль програмування. Вміти програмувати функції користувача, підпрограми на мовах Assembler та C++.	Тести, задачі, питання
2 / 4	12. Поняття структурного (структурно-алгоритмічного) програмування.	Знати поняття структурного (структурно-алгоритмічного) програмування. Розуміти принципи виклику підпрограм з основної програми. Знати прийоми введення та виведення даних із застосуванням текстового файлу.	Тести, задачі, питання
2 / 4	13. Обробка виняткових ситуацій.	Знати принципи використання блоку try ... catch. Вміти програмування блоку try усередині функції, відокремлювати обробку виняткових ситуацій. Знати прийоми перехоплення класів виняткових ситуацій. Вміти генерувати виняткові ситуації, обробляти непередбачені виняткові ситуації.	Тести, задачі, питання
2 / 4	14. Поняття структур даних. Структури. Списки. Стеки. Черги. Об'єднання.	Розуміти поняття структур даних. Вміти програмувати структур даних, списків, стеків, черг.	Тести, задачі, питання
2 / 4	15. Основи об'єктно-орієнтовного програмування.	Розуміти основи об'єктно-орієнтовного програмування. Знати структуру класу на мові C++. Знати поняття компонент, принципи об'єктно-орієнтовного програмування: інкапсуляція, наслідування, поліморфізму. Розуміти ієрархію класів. Знати поняття селектора, конструктора і деструктора, функцій-методів.	Тести, задачі, питання

Літературні джерела

1. C++ & Builder. Практикум. / Я.М. Глинський, В.Є. Анохін, В.А. Ряжська. – Львів: Деол, 2008. – 190 с.
2. Архангельский А.Я. Программирование в C++ Builder 5. / А.Я. Архангельский . – М.:ООО "БИНОМ-ПРЕСС", 2003. – 1152 с.
3. Архангельский А.Я. Приемы программирования в Borland C++. / А.Я. Архангельский. – М.:ООО "БИНОМ-ПРЕСС", 2003. – 784 с.
4. Дейтел Х. Как программировать на С/С++ / Х.Дейтел і П.Дейтел. – М.-К.:С.П., 2006. – 908 с.
5. Теорія статистики: навч. посіб. / М.К. Шапочка, О.М. Маценко. – Суми: Університетська книга, 2014. – 312 с.

6. Глушаков С. В. Программирование / С. В. Глушаков. – Харьков: «ФОЛІО», 2001. – 536 с.
7. Стивен Г., Билл И. Самоучитель Visual C++6 в примерах. – Киев: ДіаСофт, 2000. – 496 с.
8. Оберг Роберт Дж. Архитектура .NET и программирование с помощью Visual C++ / Оберг Роберт Дж., Торстейнсон Питер. :Пер.с англ. – М. :Издательский дом "Вильяме", 2002. – 656 с.
9. Тулашвілі Ю.Й. Основи прикладного програмування: Навчальний посібник / Ю. Й. Тулашвілі. – Луцьк: РВВ ЛНТУ, 2009. – 372 с.

Інформаційні ресурси в мережі Інтернет

1. Online Compiler and IDE. [Електронний ресурс] . – URL: <https://ideone.com/>.
2. Тюгашев А.А. Основы программирования. Часть I. – СПб: Университет ИТМО, 2016. – 160 с. [Електронний ресурс] . – URL: <https://books.ifmo.ru/file/pdf/1917.pdf/>

Політика оцінювання

– Політика щодо дедлайнів та перескладання:

Роботи, які здаються із порушенням термінів без поважних причин, оцінюються на нижчу оцінку (-20 балів). Перескладання модулів відбувається із дозволу деканату за наявності поважних причин (наприклад, лікарняний).

– Політика щодо академічної доброчесності:

Списування під час контрольних робіт та екзаменів заборонені (в т. ч. із використанням мобільних пристроїв).

– Політика щодо відвідування:

Відвідування занять є обов'язковим компонентом оцінювання, за яке нараховуються бали. За об'єктивних причин (наприклад, хвороба, міжнародне стажування) навчання може відбуватись в он-лайн формі (змішана форма навчання) за погодженням із керівником курсу.

Оцінювання

Остаточна оцінка за курс розраховується наступним чином:

Види оцінювання	% від остаточної оцінки
Опитування під час занять – усно	40
Модуль 1 (теми 1-6) – модульна контрольна робота	30
Модуль 2 (теми 7-15) – модульна контрольна робота	30
Екзамен (теми 1-15) – тести, завдання	60

Шкала оцінювання студентів:

ECTS	Бали	Зміст
A	90-100	відмінно
B	82-89	добре
C	74-81	добре
D	64-73	задовільно
E	60-63	достатньо
FX	35-59	незадовільно з можливістю повторного складання
	1-34	незадовільно з обов'язковим повторним курсом