МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Львівський національний університет ім. Івана Франка
Факультет прикладної математики та інформатики
Кафедра програмування

ПРОГРАМА КУРСУ

“Основи програмування”

Напрям : інформатика

Факультет : прикладної математики та інформатики

Форма навчання : денна

Виписка з навчального плану

Семестр	Кількість кредитів	Загальний обсяг (год.)	Всьогоаудит.(год.)	у тому числі (год.):			Самостійна роб.(год)	Контрольні (модульні) роботи	Курсові роботи (проекти)	Залік	Іспит
				Лекції	Лабораторні	Прак тичні					
1			90	54	36			1			+
2			68	34	34			1			+
3			72	36	36			1			+

1. АНОТАЦІЯ

Вивчаються основи програмування засобами алгоритмічної мови C++. Ознайомлення з *парадигмою процедурного програмування* передбачає освоєння вбудованих типів даних та основних інструкцій мови, конструювання простих регулярних типів і реалізацію алгоритмів за допомогою функцій. У рамках *об’єктно-орієнтованої парадигми* вивчається конструювання нових типів за допомогою концепції класу; на практиці розглядаються шаблони проектування для задання взаємозв’язків між типами і об'єктами. *Парадигма узагальненого програмування* викладається на прикладі використання контейнерних класів та алгоритмів стандартної бібліотеки шаблонів.

2. ЗМІСТ ПРОГРАМИ

1. Типи даних в С++. С++ як сильно типізована мова програмування. Система типів. Внутрішнє представлення даних. Оголошення та визначення змінних та констант. Поняття про ініціалізацію змінних.

2. Вирази. Класифікація операторів. Константні вирази. Поняття про первинні вирази. Оператори присвоєння, L-вирази. Поняття про еквівалентність і перетворення типів. Використання виразів з потоковими об’єктами для виведення даних на консоль.

3. Інструкції. Оголошення як інструкція С++. Пуста інструкція, блок, інструкція-вираз, if, switch, for, while, do, continue, break, goto, return.

4. Оголошення та визначення. Відмінності оголошення та визначення. Правило одного визначення. Тип – сутність з атрибутами.

5. Масиви і вказівники. Масиви як регулярні типи. Вказівники – низькорівневий засіб С++. Алгоритми виконання new і delete. Спорідненість вказівників та масивів. Посилання як константний вказівник, що автоматично розіменовується.

6. Функції. Функція як основна програмна одиниця, метод декомпозиції. Поняття про прототип функції. Особливості оголошення функції, перевантаження. Виклик функції.

7. Класи – основний засіб визначення типів. Визначення типів за допомогою класів і структур**.** Протокол класу як засіб логічного об’єднання даних і методів, інкапсуляція. Об’єкти як екземпляри класу. Оператори доступу до членів класу. Поля даних: характеристика стану об’єкта, специфікація доступу. Дані об'єкта і дані класу. Види методів. Визначення методів. Вказівник this. Конструктор як спеціальний метод класу. Перевантаження конструкторів.

8. Перевизначення операторів. Синтаксис операторних функцій; загальні правила перевизначення. Відмінність використання перевизначених операторних функцій-методів і глобальних функцій. Особливості перетворення типів аргументів в операторних функціях, неявне перетворення. Оператор присвоєння. Глобальне перевантаження операторів. Дружні функції. Задання операторів обміну даними з потоками введення/виведення.

9. Деструктор класу. Звільнення пам'яті деструктором. Загальні правила реалізації деструкторів.

10. Перетворення типів. Випадки явного і неявного перетворення. Неявне перетворення конструктором; explicit конструктор. Оператори перетворення (приведення) типу.

11. Ознайомлення з функціональністю класу string. Typedef – задання синоніму для оголошення. Поняття про basic_string і string. Конструктори класу string,

способи утворення рядка символів. Оператори, екстрактори та інсертори. Функції getline і swap. Методи класу basic_string.

12. Константні конструкції C++. Доцільність використання констант. Константи вбудованих типів. Константи агрегатів. Фізична і логічна константність, mutable-об’єкти. volatile конструкції.

13. Механізм контролю назв. Використання static-конструкцій для внутрішнього зв'язування і статичного розміщення. Клас як область видимості; особливості використання статичних членів. Статичні глобальні об'єкти. Вкладені і локальні класи. Простори назв: оператор області видимості, using- директива і оголошення.

14. Наслідування як механізм повторного використання коду. Синтаксис похідних класів. Ієрархія областей видимості. Особливості реалізації конструкторів похідних класів.

15. Поліморфізм та віртуальні методи. Пояняття про раннє (статичне) та пізнє (динамічне) зв’язування. Поліморфний кластер. Поняття про способи реалізації поліморфізму; таблиця віртуальних методів, вказівник на таблицю віртуальних методів. Основні правила визначення віртуальних методів. Абстрактні класи; чисті віртуальні методи. Віртуальні деструктори.

16. Взаємозв’язки між класами. Наслідування та композиція, особливості використання вкладених і базових підоб’єктів. Засоби модифікації інтерфейсу базового класу. Перевизначення віртуальних функцій і перевантаження методів. Приклади розробки множини класів для абстрактних динамічних структур даних – списків і стека. UML-діаграми для моделювання ієрархій типів.

17. Множинне наслідування. Синтаксис класів у випадку множинного наслідування. Порядок виклику конструкторів та деструкторів базових класів. Віртуальні базові класи: порядок ініціалізації, порядок виклику конструкторів, множинне використання базового класу. Особливості множинного наслідування.

18. Винятки як системний підхід до обробки помилок. Оператори генерування винятків у try-блоках. Перехоплення винятків та catch блоки. Обробіток виняткових ситуацій в конструкторах. Специфікація інтерфейсу функцій щодо винятків. Стандартні винятки C++.

19. Основи параметризованих функцій і класів. Поняття про параметризовані функції. Вивід аргументів. Параметри шаблонів. Перевантаження функцій і шаблони. Синтаксис параметризованого класу. Проблема множини значень параметрів шаблону. Параметри шаблону, які не є типами. Основи роботи з шаблонами; поняття про інстанціювання. Явне інстанціювання. Шаблонні параметри шаблонів. Аргументи шаблона за замовчуванням. Шаблони і дружні типи та функції. Наслідування і шаблони класів. Пошук назв. Види інстанціювання.

20. Ієрархія потокових шаблонів. Огляд ієрархії потоків. Стандартні потоки. Форматування засобами класу ios. Стандартні маніпулятори. Файлові потоки Потоки для зберігання та обробки даних в оперативній пам'яті; засоби буферизації; методи введення/виведення та форматування символьних рядків.

21. Шаблони і проектування програм. Поліморфні можливості шаблонів: динамічний та статичний поліморфізм. Класи властивостей і стратегії. Поняття про метапрограмування.

22. Узагальнене програмування на основі бібліотеки стандартних шаблонів (STL). Концепція стандартної бібліотеки. Загальні властивості та основні операції над послідовними контейнерами. Загальні властивості та основні операції над асоціативними контейнерами. Категорії ітераторів. Допоміжні функції ітераторів. Ітераторні адаптери. Патерн Adapter.

23. Алгоритми і об'єкти-функції. Класифікація і загальний огляд алгоритмів. Концепція об'єктів-функцій та її реалізація. Використання з алгоритмами для обробки контейнерів. Предикати. Функціональні адаптери.

24. Патерни проектування. Класифікація патернів. Приклади патернів створення об'єктів (Singleton, Factory Method), особливості реалізації типами C++. Структурні патерни (Adapter, Bridge). Приклади патернів поведінки (Command, Iterator, Strategy).

ОСНОВНА ЛІТЕРАТУРА

1. **Страуструп Б.** Язык программирования C++. Специальное издание(3-е издание), 1999.
2. **Эккель Б.** Философия C++, т.1,2. 2-е издание, 2004.
3. **Макконнелл С.** Совершенный код.-2005.
4. **Саттер Г., Александреску А.** Стандарты программирования на C++.-2005.
5. **Кениг Э., Му Б.** Эффективное программирование на C++. –2002.
6. **Страуструп Б.** Дизайн и эволюция C++.-2000.

Автори програми:

Клакович Л.М., канд.фіз.-мат.наук, доцент кафедри програмування
Музичук А.О., канд.фіз.-мат.наук, доцент кафедри програмування