# МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

**Національний університет “Львівська політехніка”**



**Інститут післядипломної освіти**

**Кафедра програмного забезпечення систем**

**ЗВІТ**

**Про виконання лабораторної роботи №1**

**«Основні поняття мови С++»**

**з дисципліни «Основи програмування»**

Виконав:

слухач групи ПЗС-11

Гринчук Тарас

Прийняв:

доц. Макар В.М.

« »\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2013 р.

∑ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

ЛЬВІВ – 2013

**Тема роботи**: Основні поняття мови С++

**Мета роботи:** навчитися програмувати на мові С++ найпростіші лінійні алгоритми із застосуванням потокового введення вхідних даних та виведення результатів.

## 1. ТЕОРЕТИЧНІ ВІДОМОСТІ

1. **Ідентифікаторами** називаються усі імена, які використовуються в програмі для означення різних об’єктів і конструкцій. Усі ідентифікатори мови С++ можна поділити на три групи:
   * **Службові (зарезервовані) слова** мови
   * **Бібліотечні ідентифікатори**
   * **Ідентифікатори користувача**

Основні правила запису ідентифікаторів:

* Ідентифікатор складається з букв, цифр і знаку підкреслення (до складу ідентифікатора не може входити будь-який інший спеціальний символ).
* Першою повинна бути тільки буква. Не рекомендується починати ідентифікатори зі знаку підкреслення, оскільки багато бібліотечних об’єктів та функцій починаються саме з цього знаку.
* У мові С++ розрізняється регістр букв, тобто, наприклад, Х і х - це два різні ідентифікатори.
* Ідентифікатори користувача не повинні співпадати з службовими словами та біблиотечними ідентифікаторами.

1. **Змінні** – це величини, які можуть змінюватися в процесі виконання програми, а **константи** – це величини, які не змінюють свого значення протягом усього часу виконання програми.
2. Числа дійсного типу мають дві форми запису:
   * з фіксованою крапкою – у вигляді цілої та дробової частини, розділеної крапкою: 25.34, 3.1415;
   * з плаваючою крапкою – у вигляді мантиси та порядку, розділених буквою Е: 0.31415Е1.
3. До простих типів мови С++ відносяться:
   * **сhar** - одиничний байт, що містить один символ;
   * **int** - ціле число;
   * **float** - число з плаваючою крапкою одиничної точності;
   * **double** - число з плаваючою крапкою подвійної тосності.

Для розширення базових типів використовуються *кваліфікатори*:

1. ***short -*** короткий;
2. **long** - довгий;

Кваліфікатор:

1. ***signed*** - із знаком;
2. ***unsigned*** - без знака (застосовуються до типів int i та char);
3. Коментарі використовуються для того, щоб документувати програму і полегшити її розуміння. Коментарі допомагають іншим людям читати і розуміти Вашу програму. Коментар, який починається з символа // називається **однострічковим** коментарем, тому, що він закінчується в кінці поточного рядка. **Багатострічковий** коментар в С++ повинен розміщатися між парою символів /\* та \*/.
4. Якщо в програмі використовуються тексти інших бібліотек, потрібно перед написанням коду основної програми, директивою #include включити в текст програми вміст відповідного файлу (напр. #include <iostream>). Файл iostream.h - прийнято називати  *заголовочним файлом.* Будь-яка С++ програма складається з однієї чи декількох функцій, одна з яких повинна обов’язково називатися main. Функція main називається  *головною функцією* і вона є точкою входу в програму. Це означає, що кожна програма на С++ починає виконуватися з функції main, навіть якщо ця функція не є першою в тексті програми. Ліва фігурна дужка { починає  *тіло* кожної функції. Відповідно, права фігурна дужка } повинна завершувати кожну функцію. Тіло функції може містити довільну кількість операторів та викликів інших функцій. Приклад найпростішої програми:

#include <iostream>

void main() {

using namespace std;

cout<<”Hello, world !”;

}

Рядок

using namespace std;

містить команду  **using namespace** , яка не вимагається стандартом мови С++ і є специфічною особливістю середовища MS Visual C++ 2010. Ця команда просто вказує на те, що в нашій програмі буде використовуватися простір імен  **std**, в якому описані багато стандартних об’єктів, наприклад об’єкт **cout.**

**Приклад програми**

#include <iostream>

#include <conio.h> //підключення заголовочних файлів

int main() { //початок головної функції

using namespace std; //використання простору імен std

cout<<"My first C++ program ! \n"; //виведення тексту\_

\_getch(); //затримка виконання до натиску

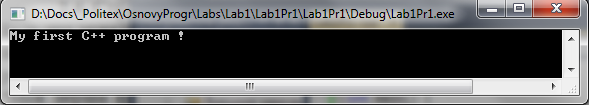
//будь-якої клавіші

return 0; //функція main повинна повернути

//результат з типом значення int

} //кінець головної функції main

**Результат виконання:**

****

(Рис. 1.1)

1. Виведення на екран в С++ виконується, за допомогою команди cout<<”Текст повідомлення”;

Ця команда має вигляд оператора виведення, який складається з ідентифікатора об’єкта *стандартного потоку виведення*  **cout**, операції << , рядка ”Текст повідомлення” та символа крапки з комою ; , який є  *ознакою завершення оператора* . Кожний оператор в С++ повинен закінчуватися цим символом. Будь-яке введення/виведення в С++ виконується над *потоками* символів. Таким чином, виконання даного оператора виведення полягає у передачі потоку символів ”Текст повідомлення” об’єкту стандартного потоку виведення cout, який асоціюється з екраном монітора. Операція << називається  *помістити в потік* . Ця операція має два  *операнди* (те над чим виконується операція): зліва - об’єкт cout, справа – рядок символів ( *стрічкова константа* або *літерал*).

1. Якщо виконати команду cout<<”Текст повідомлення\n”, то можна побачити, що символи \n з даного літералу не виводяться на екран. Символ \ ( *обернений слеш* ) називається  *символом переходу* або  *escape-символом*. Він позначає собою початок так званої *керуючої послідовності* або *escape-послідовності*, яка задає виведення певного спеціального символу. У нашому випадку керуюча послідовність  **\n** означає  *початок нового рядка* і викликає переміщення курсора (тобто індикатора поточної позиції на екрані) на початок наступного рядка на екрані. Деякі інші, найчастіше вживані escape-послідовності:
   * \n - Новий рядок, курсор встановлюється на початок нового рядка;
   * \t - Горизонтальна табуляція, курсор встановлюється на наступну позицію табуляції;
   * \r - Повернення каретки, курсор встановлюється на початок поточного рядка;
   * \\ - Друк символа \
   * \” - Друк символа “.
2. Елементарні операції в С++ можна поділити на такі групи:
   * арифметичні операції;
   * операції порівняння;
   * операція присвоєння.

*Арифметичні операції* : додавання +, віднімання -, множення \*, ділення /, залишок від ділення % виконуються над операндами цілого або дійсного типу. Перераховані вище операції є  *бінарними*, оскільки вони мають два  *операнди* . Крім них у мові С++ є специфічні  *унарні* операції (тобто такі які мають лише один операнд)  *інкременту* ++ та  *декрименту* --. Інкрементна операція ++ збільшує значення свого операнда на 1. Оператор n++; аналогічний такому оператору: n=n+1; Декриментна операція -- віднімає 1 від поточного значення свого операнда.

**Приклад програми**

#include <iostream>

#include <conio.h> //підключення заголовочних файлів

void main() { //початок головної функції

using namespace std; //використання простору імен std

int a=1, b=1; //присвоєння початкових значень

int aplus, plusb; //оголошення змінних

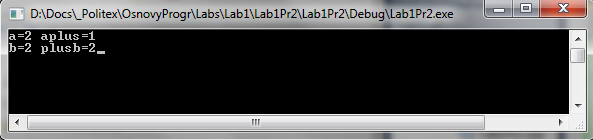
aplus=a++; plusb=++b;//обчислення значень

cout<<"a="<<a<<" aplus="<<aplus<<endl; cout<<"b="<<b<<" plusb="<<plusb; //виведення результату на екран

\_getch(); //затримка до натиску клавіші

} //кінець головної функції

**Результат виконання:**



(Рис. 1.2)

Значення а збільшилось на 1 після того як виконалась операція присвоєння. Значення b спочатку збільшилось на 1, а тоді виконалась операція присвоєння.

1. Розрізняють дві форми запису операцій інкременту та декрименту:
   * *префіксну* ++n - змінна n збільшується на 1 до того, як використовується у виразі;
   * *постфіксну* n++ - змінна n збільшується на 1 після того, як її значення буде використано у виразі.
2. При використанні операцій важливе значення відіграють пріоритет (старшинство) та асоціативність (порядок виконання) операцій. Групування за старшинством (в порядку спадання пріоритету) наведено у таблиці (таб 1.1):

|  |  |
| --- | --- |
| Операція | Асоціативність |
| () | Зліва направо |
| ++ -- | Справа наліво |
| \* / % | Зліва направо |
| + - | Зліва направо |
| < <= > >= | Зліва направо |
| == != | Зліва направо |
| << >> | Зліва направо |
| = | Справа наліво |

(Таб 1.1)

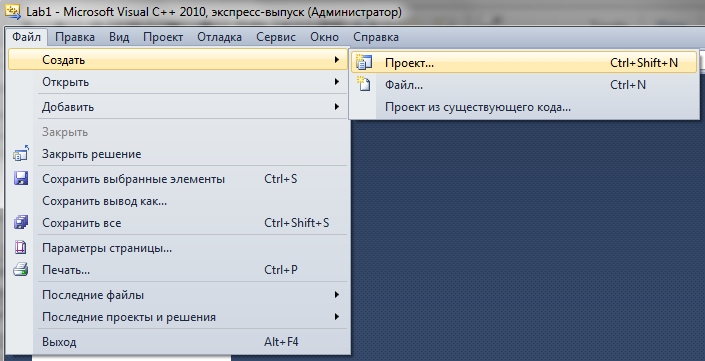
1. Програми на С++ містять частини, які називаються  *класами* і  *функціями*. Бібліотеки стандартних функцій містять багато наборів вже існуючих класів і функцій з бібліотек класів С++ і бібліотеки стандартних функцій ANSI. Використання функцій стандартної бібліотеки замість написання власних версій тих же функцій може підвищити ефективність програм, оскільки ці функції написані спеціально з урахуванням ефективності їх виконання.

## ХІД РОБОТИ

**Індивідуальне завдання (варіант 5).** Написати програму для обчислення за арифметичного виразу:

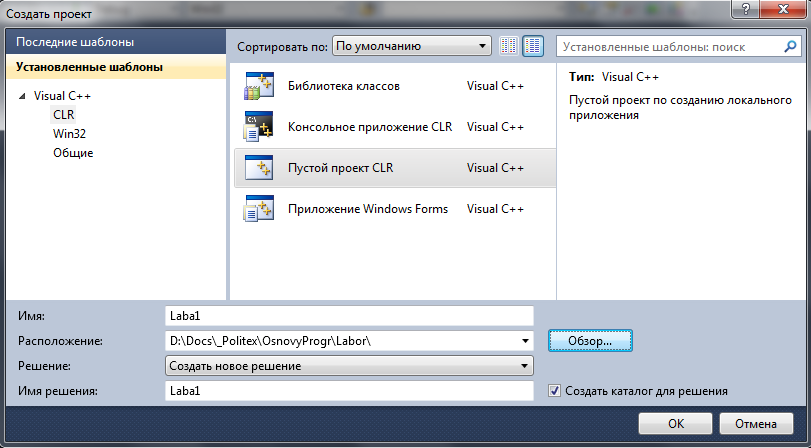
Вважати, що X, Y – змінні, значення яких слід вводити з клавіатури, всі інші величини виразу описати як константи.

1. Запускаємо середовище **Microsoft Visual C++ 2010**. Створюємо новий проект за допомогою команди головного меню: *«Файл/Создать/Проект…»* (рис. 2.1).



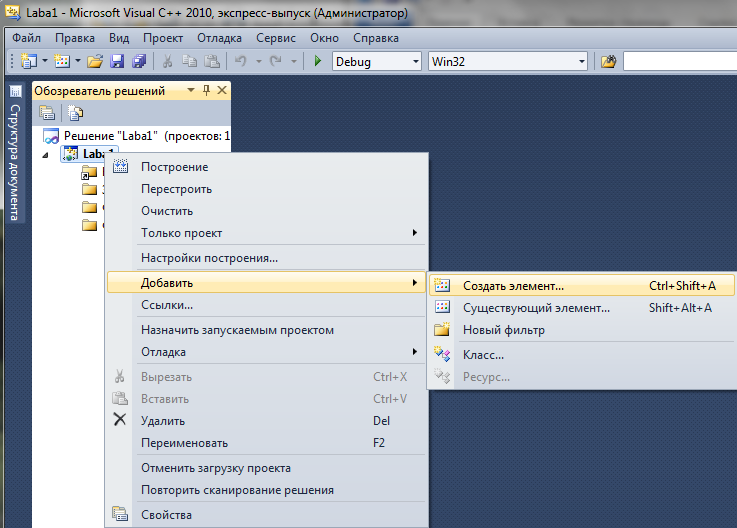
(Рис 2.1)

1. Вибираємо категорію *«Пустой проект CLR»,* задаємо місцезнаходження проекту на диску та його назву (рис 2.2).



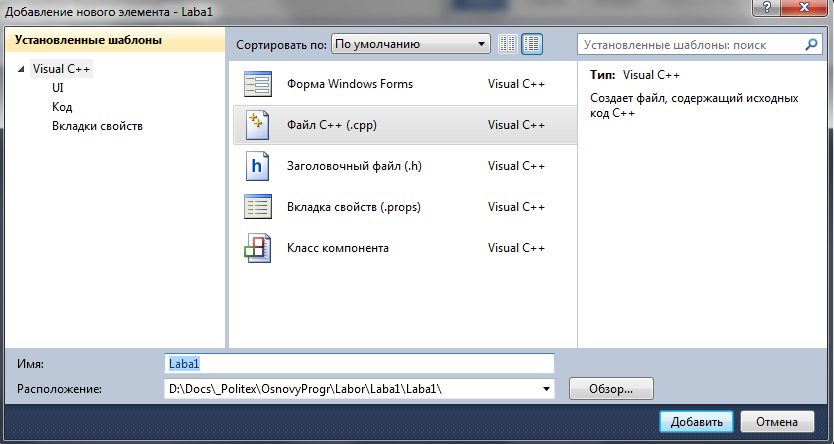
(Рис 2.2)

1. Натиском правої кнопки миші, по дереву проекту, викликаємо контекстне меню і виконуємо команду *«Добавить/Создать элемент…»* (рис 2.3).



(Рис 2.3)

1. Задаємо тип елемента: *«Файл С++ (.срр)»* та назву цього файлу (рис 2.4).



(Рис 2.4)

1. Набираємо у вікні редактора текст програми:

#include <iostream>

#include <conio.h>

#include <math.h> //підключення заголовочних файлів

void main() { //початок головної функції

using namespace std; //використання простору імен std

float x; //визначення змінної х

float const a=3, p=6, b=2;//визначення констант a,p,b

cout<<"x: "; //запит до користувача ввести х

cin>>x; //отримання з клавіатури х

float c = x/a-1/(a\*p)\*log(a+b\*exp(p\*x)); //обчислення виразу

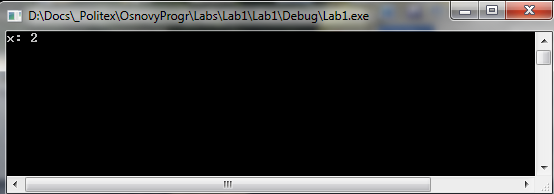
cout<<"c="<<c<<endl; //виведення результату на екран

\_getch(); //затримка програми до натиску

//довільної клавіші

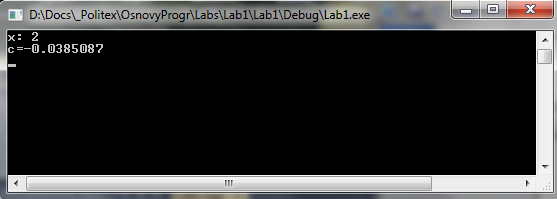
} //кінець головної функції

1. Зберігаємо зміни *(Файл/Сохранить все),* запускаємо програму на виконання *(Отладка/Начать отладку).* Вводимо значення змінної «х» рівним 2. (рис. 2.5)



(Рис 2.5)

1. Отримуємо результат обчислення виразу с=-0.0385087 (рис. 2.6).



(Рис 2.6)

1. Натиском будь-якої клавіші, завершуємо роботу програми.

## ВИСНОВКИ

## На даній лабораторній роботі я отримав базові знання мови програмування С++, а саме: синтаксису мови, використання функцій вводу/виведення на екран, обчисленню математичних виразів, підключенню зовнішніх бібліотек функцій, а також створенню програм, за допомогою середовища Microsoft Visual C++ 2010.