**Практична робота №6. Написання програми математичних розрахунків з введенням вхідних даних та виведенням результату обчислень**

**Запитання**

1. Як забезпечується організація введення даних на С++?
2. Як здійснюється виведення результатів на С++?
3. Як реалізується потокове введення даних «**сіn>>**»? Наведiть приклади.
4. Охарактеризуйте основні аспекти використання потокового виведення даних «**соut<<**».
5. Для чого слугують заголовні файли?
6. Як компілятор шукає розташування заголовних файлів?
7. Які маніпулятори використовуються для завдання довжини поля, дробової частини числа?
8. Який заголовний файл використовується для маніпуляторів завдання довжини поля, дробової частини числа?

**Мета:** набуття навичок створення С++-програм із використанням управляючих структур розгалуження (вибору) для виконання розрахунків

**Завдання.**

Обчислити і вивести на екран значення функції *y*.

1. 
2. 
3. 

Результати надсилати на електронну адресу викладача [**t.i.lumpova@gmail.com**](mailto:t.i.lumpova@gmail.com)у вигляді cpp-файлу з іменем у форматі

**<Номер групи><Номер лабораторної><Прізвище англійською>**

Наприклад, 31-01Ivanov.cpp.

Іншим рішенням є надсилання поштою Звіт зі скріншотом програми в C++Shell, вказавши в темі листа, номер групи прізвище студента та номер ПР.

В темі листа вказати, номер групи, прізвище студента та номер ПР як "ПР№5".

**Строк відсилки ЛР для МНТ/ЕТ-41 03.03.2024**

Всі запитання, що виникнуть, надсилайте на електронну адресу викладача.

Всі запитання, що виникнуть, надсилайте на електронну адресу викладача, тему в заголовку листа записати

**ОРІ-Запитання-<Номер групи>-<Прізвище англійською>**.

**Теоретична частина**

У програмах С++ може бути передбачений перехід з однієї частини програми до іншої в залежності від виконання або невиконання деякої умови. Оператори, що реалізують подібні переходи, називають *умовними операторами*. Умовні оператори поділяються на дві основні категорії: повторення (цикли) та розгалуження (вибору).

В С + + існує декілька типів розгалужень, найбільш важливим з яких є розгалуження if ... else, що здійснює вибір між двома альтернативами. Для вибору однієї з багатьох альтернатив використовують розгалуження switch … case, дія якого визначається набором значень керуючої змінної.

**Умовний оператор if**

Оператор if є найбільш простим з операторів розгалуження. Його синтаксис такий:

**if ( умова )**

**{  
 інструкція;  
 ……………** // Тіло розгалуження з кількох операторів

**інструкція;  
 }**

За ключовим словом if слідує *умова* розгалуження, поміщена в круглі дужки. Умова – це вираз, від істинності чи хибності якого залежить виконання або невиконання певної частини програми. Якщо умова істинна, *інструкції* виконуються, в іншому випадку – ні.

Тіло розгалуження може складатися як з одного оператора, так і з кількох операторів (інструкцій), укладених у фігурні дужки.

**Oператор if ... else**

Оператор if дозволяє здійснювати певну дію лише у тому випадку, коли виконується деяка умова. Якщо ж умова не виконується, то й ніякої дії не виконується. Коли необхідно зробити одну дію у разі виконання умови, а іншу дію у разі невиконання цієї умови, то використовують розгалуження if ... else. Воно складається з оператора if, за яким слідує блок інструкцій, і ключового слова else, за яким слідує ще один блок інструкцій.

Синтаксис розгалуження if ... else такий:

**if ( умова )**

**{  
 інструкція;  
 ……………** // Тіло if з кількох операторів

**інструкція;  
 }**

**else**

**{  
 інструкція;  
 ……………** // Тіло else з кількох операторів

**інструкція;  
 }**

Для множинного вибору можна використовувати *вкладені* структури if ... else, розміщуючи одну структуру if ... else всередину іншої.

**Умовний оператор**

Замість оператора if…else у С++-програмі може використовуватися у*мовний оператор.* Умовний оператор**?:**  дає можливість створювати прості однорядкові умовні вирази, в яких виконується одна з двох дій залежно від значення умови.

Синтаксис умовного оператора такий:

**умова ? оператор1 : оператор2;**

Якщо *умова* приймає ненульове значення, то виконується *оператор1*, інакше – *оператор2.*

**Оператор switch**

Для організації структур із множинним вибором у мові С++ передбачено використання оператору **switch**. Ця структура складається з ряду міток **case** і необов’язкового блока **default**.

Синтаксис структури switch наступний:

**switch (вираз умови)**

**{**

**case константа 1:**

**інструкція;**

**. . . . .**

**інструкція;**

**break;** //вихід із switch

**case константа 2:**

**інструкція;**

**. . . . .**

**інструкція;**

**break;** //вихід із switch

**case константа 3:**

**інструкція;**

**. . . . .**

**інструкція;**

**break;** //вихід із switch

**default:**

**інструкція;**

**. . . . .** // оператори за замовчуванням

**інструкція;**

**}**

За ключовим словом switch в круглих дужках слідує *вираз умови*. Цим виразом може бути будь-який, допустимий у мові С++ вираз, значення якого повинно бути цілим.

Тіло оператора smitch складається з декількох фрагментів, що складаються з ключового слова case і *константного виразу*, з яким порівнюється вираз умови. У якості константного виразу використовуються цілі або символьні константи. Всі константні вирази в операторі switch повинні бути унікальні.

Крім операторів, помічених ключовим словом case, може бути, але обов'язково один, фрагмент, позначений ключовим словом default. Інструкції, відповідні цій мітці, виконуються, якщо вираз умови не відповідає жодній з міток case. Мітка default є аналогом частини else інструкції if-else.

В абсолютній більшості випадків за кожною міткою case повинен слідувати відповідний оператор break. Використання оператора break дозволяє в необхідний момент перервати виконання операторів у тілі switch і передати управління першому після структури switch оператору, тобто виконати вихід зі структури. Якщо немає оператора break, то відбувається виконання всіх наступних виразів до фігурної дужки, що завершує структуру switch. Таким чином, оператор break реалізує схему альтернативного вибору.

Зверніть увагу! В операторі switch не потрібно укладати послідовність операторів у фігурні дужки.

**Операції відношень**

Вираз умови розгалуження записують за допомогою особливого типу операцій, що їх називають *операціями відношень*. Операція відношення порівнює між собою два значення. Результатом порівняння є значення *істина* або *брехня*.

В С++ визначено повний набір операторів відношень, які можуть бути використані в умовному виразі:

**== Дорівнює**

**!= Не дорівнює**

**> Більше**

**< Менше**

**>= Більше або дорівнює**

**<= Менше або дорівнює**

**Логічні операції**

С++ надає також три *логічні операції*, які можна використовувати для формування складних умов шляхом комбінування простих.

Логічними операціями є:

**&& - Логічне І**

**|| - Логічне АБО**

**! - Логічне НЕ (логічне заперечення)**

Для оцінки дії логічних операцій використовують таблиці істинності.

Таблиця істинності операції &&

|  |  |  |
| --- | --- | --- |
| Вираз 1 | Вираз 2 | Вираз 1 && Вираз 2 |
| false | false | false |
| false | true | false |
| true | false | false |
| true | true | true |

Таблиця істинності операції ||

|  |  |  |
| --- | --- | --- |
| Вираз 1 | Вираз 2 | Вираз 1 || Вираз 2 |
| false | false | false |
| false | true | true |
| true | false | true |
| true | true | true |

Дія операції логічного заперечення **(**!) полягає в тім, що вона змінює значення свого операнда на протилежне: якщо операнд мав істинне значення, то після застосування операції ! він стає хибним, і навпаки.

Операції відношень мають вищий пріоритет, ніж логічні операції. А операція логічного І (&&) має більш високий пріоритет, ніж операція логічного АБО (||).

**Приклад програми**

**Завдання**: Обчислити і вивести на екран значення функції :

*y*=

**#include <iostream>**

**#include <conio.h>**

**using namespace std;**

**void main()**

**{**

**float x, y;**

**cout << " Enter number: ";**

**cin >> x;**

**if ( x > =0 )**

**y=x;**

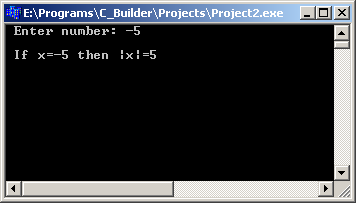
**else**

**y=-x;**

**cout << “\n If x=”<<x<<” then |x|=”<<y<<endl;**

**getch();**

**}**



**Контрольні питання**

1. Які оператори називають умовними операторами?
2. Назвітьумовні оператори С++.
3. Назвіть типи розгалужень С + +.
4. Що таке повна та неповна форма розгалуження?
5. Наведіть загальний синтаксис структур вибору С++.
6. Які операції називають операціями відношень?
7. Назвіть оператори відношень, визначені у С++.
8. Яке призначення логічних операцій?
9. Назвіть логічні операції С++.