**Введення, виведення інформації. Оператор присвоєння.**

Розглянемо способи вводу та виводу інформації на мові програмування С++. В даній темі буде розглянуто потоковий ввід та вивід інформації. Існують і інші способи введення та виведення інформації, в цій темі вони розглянуті не будуть.

Існує 3 стандартних потоки інформації, які використовує кожна програма – потік вводу інформації, потік виведення інформації та потік виведення помилок. В нашому випадку клавіатура відповідає потоку вводу, консоль відповідає потоку виводу і потоку виведення помилок.

В більш складніших програмних продуктах стандартні потоки можуть бути замінені на інші. Наприклад стандартний потік помилок може бути замінений на файл. Помилки не будуть відображатись в консолі з іншими програмними повідомленнями і їх не можливо буде загубити, а також спутати зі звичайними повідомленнями програми.

Об’єкти які ми будемо використовувати мають назву «cin» (потік вводу), «cout» (потік виводу), «cerr» (потік помилок) з простору імен «std». Так як в нашому випадку потік виводу інформації та потік виводу помилок виводять інформацію в консоль - буде розглянуто тільки потік виводу інформації.

Щоб скористатися об’єктами для вводу та виводу в С++ необхідно спочатку підключити бібліотеку «iostream». (input output stream)

Робиться це командою

#include <iostream>

Попередження: файл бібліотеки повинен бути підключеним ДО використання функцій чи об’єктів бо станеться помилка.

Як ви вже знаєте програма на мові С++ бере свій початок в функції з назвою «main», і компілятор відмовить вам з помилкою, якщо не знайде в коді цю функцію. Тому саме в ній ми будемо писати свій код.

int main() {

std::cout << " Hello, world!\n";

}

Після компіляції і запуску програма виведе повідомлення «Hello, world!». Але не зручно звертатись до об’єкту «cout» через повний шлях. Тому перед фунцією «main» ми можемо вказати простір імен, який слід використовувати програмі. Це нам дасть змогу звертатись до об’єкта напряму, а не через повний шлях. Наші об’єкти вводу та виводу інформації лежать в просторі імен «std», тому давайте вкажемо програмі використовувати його, щоб не писати повний шлях до об’єкту. Нижче повний код програми, яка виводить «Hello, world!».

#include <iostream>

using namespace std;

int main() {

cout << " Hello, world!\n";

}

Якщо виведення інформації виводить інформацію на консоль і не зберігає її, то введенні дані потрібно десь зберігати. Ви вже знайомі з поняттям змінної. Образно кажучи це комірка, в яку можна щось покласти. С++ строго типізований язик програмування, тому при створенні змінної необхідно вказувати який тип даних буде зберігатися в змінній. Це може бути ціле число або символ. Типів даних в С++ більше, не будемо на цьому зупинятись. В прикладі нижче ми будемо використовувати тип для цілих чисел – int (скорочення від integer). Щоб створити змінну в С++ достатньо вказати тип змінної та її назву:

int x;

Є певні обмеження при створенні назв змінних – вони можуть починатись лише с букви або з символа підкреслювання «\_».

Ми змінну створили, але в ній зараз нічого нема! Давайте використаємо оператор ПРИСВОЄННЯ «=», щоб надати змінній конкретне значення, наприклад 3:

x = 3;

Операція присвоєння відбувається з ПРАВА НА ЛІВО. Словами цю дію можна описати як «В змінну Х покласти значення 3». Давайте перевіримо чи в нас це вдалось використавши вивід інформації з прикладу вище. Код для перевірки:

#include <iostream>

using namespace std;

int main() {

int x;

x = 3;

cout << " X = " << x;

}

Також можна відразу присвоїти значення змінній при її створенні.

#include <iostream>

using namespace std;

int main() {

int x = 3;

cout << " X = " << x;

}

За допомогою «<<» мы можем передати на потік виводу декілька аргументів. В прикладах вище передавалося 2 аргументи - рядок " X = ", а також значення «х».

Створивши і записавши значення в змінну ніщо нам не заважає при необхідності змінити значення змінної. При цьому старе значення заміняється новим:

#include <iostream>

using namespace std;

int main() {

int x = 3;

cout << " X = " << x;

x = 2;

cout << " X = " << x;

}

Змінна може зберігати результат арифметичних операцій над числами або над іншими змінними

#include <iostream>

using namespace std;

int main() {

int x;

x = 2 + 2;

cout << " X = " << x;

int y = x \* 2;

cout << " Y = " << y;

}

В даному випадку ми помістили результат додавання двох чисел в змінну «х». Потім при створенні змінної «у» відразу присвоїли їй результат арифметичної операції множення, де значення «х» - 4 було помножене на 2, а результат 8 був присвоєний в змінній «y».

Також можливо присвоїти змінній значення іншої змінної:

#include <iostream>

using namespace std;

int main() {

int x;

x = 2 + 2;

cout << " X = " << x;

int y = x ;

cout << " Y = " << y;

int z = x ;

cout << " Z = " <<z;

}

В прикладі вище змінна "у" отримала значення змінної "х" - 4. Змінна "z" також отримала значення змінної "х". При цьому значення змінної "х" зберігається, а саме значення копіюється в іншу змінну.

Давайте розберемо ще один приклад, в якому ми поміняємо значення змінних. Значення змінної "х" перейде в змінну "у". А значення змінної "у" перейде у змінну "х".

При присвоєнні значення однієї змінної іншій, інша буде перезаписана новим значенням. Щоб не втратити значення змінної при присвоєнні нового значення нам потрібно це значення десь зберегти. Використаємо допоміжну змінну, в яку ми збережемо значення "х", щоб не втратити його. Потім ми присвоїмо значення змінної "у" в змінну "х". Значення "х" стане рівне значенню "у". Потім присвоїмо змінній "у" минуле значення "х", яке ми зберегли в допоміжній змінній. Код:

#include <iostream>

using namespace std;

int main() {

int x = 2;

int y = 5;

int temp; // допоміжна змінна

temp = x; //присвоюємо змінній temp значення х, тепер вона дорівнює 2

x = y;//присвоюємо змінній х значення у, тепер х дорівнює 5

y = temp; //присвоюємо змінній у значення змінної temp, яка зберігала початкове значення х, тепер "у" дорівнює 2

cout << " x= " << x<< " y = " << y;

}

Перейдемо до операцій вводу. Створимо змінну, в котрій ми збережемо значення введене з клавіатури. Оскільки тип змінної у нас int, необхідно буде ввести число

#include <iostream>

using namespace std;

int main() {

int x;

cout << " input x = ";

cin >> x;

int y = x \* 2;

cout << " y = " << y;

}

Тут ми створюємо змінну,, але вона без значення. Потім виводимо підказку користувачу, що необхідно ввести «х» за допомогою виводу інформації. Використовуючи потік «cin» запросимо у користувача ввести дані з клавіатури, а символи «>> x» вказують на те, що результат буде збережено в змінну «х». Потім над введеним значенням виконується операція множення на 2 і виведення результату в консоль.

Також в потік «cin» за допомогою оператору «>>» можна передати декілька змінних, в які буде збережено введене з клавіатури значення. В випадку, якщо запрошується декілька значень, необхідно буде підтвердити ввід значення кнопкою «Enter». Код програми нижче попросить у користувача ввести 2 числа, які потім перемножує, зберігає в змінну «а», і виводить на консоль. При цьому значення самих «х» та «у» не змінились, вони мають ті ж самі значення, які були введені з клавіатури. Їх значення було використане в арифметичній операції добутку, а замі змінні не змінили свого значення.

#include <iostream>

using namespace std;

int main() {

int x,y;

cout << " input 2 number: ";

cin >> x >> y;

int a = x \* y;

cout << " " << x << " \* " << y << " = " << a;

}