Міністерство освіти і науки України

Національний університет « Львівська політехніка »

Інститут телекомунікації, радіоелектроніки та електронної техніки

кафедра « Радіоелектронні пристрої та системи »

З дисципліни « Програмування частина 2»

Лабораторна робота №18

«Оператори циклу»

Мета роботи: ознайомитися з поняттями структури та об’єднання даних , навчитися їх використовувати у процесі програмування.

Підготував

ст.групи АП-11

Василюк Ростислав

Прийняв:

Чайковський І.Б.

Львів 2024р

Теоретичні відомості

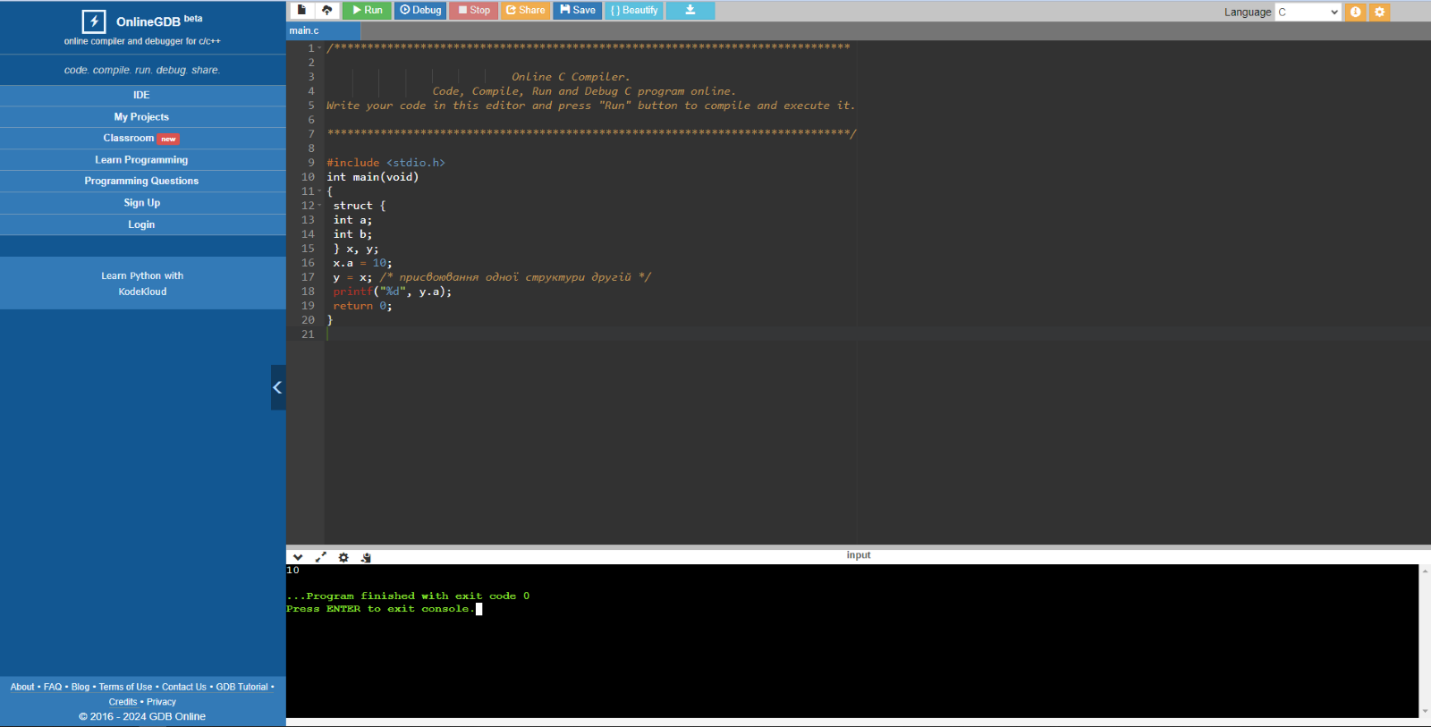
Структура – це сукупність змінних, об'єднаних під одним ім'ям. За допомогою структур зручно розміщувати в суміжних полях пов'язані між собою елементи інформації.

Хід роботи:

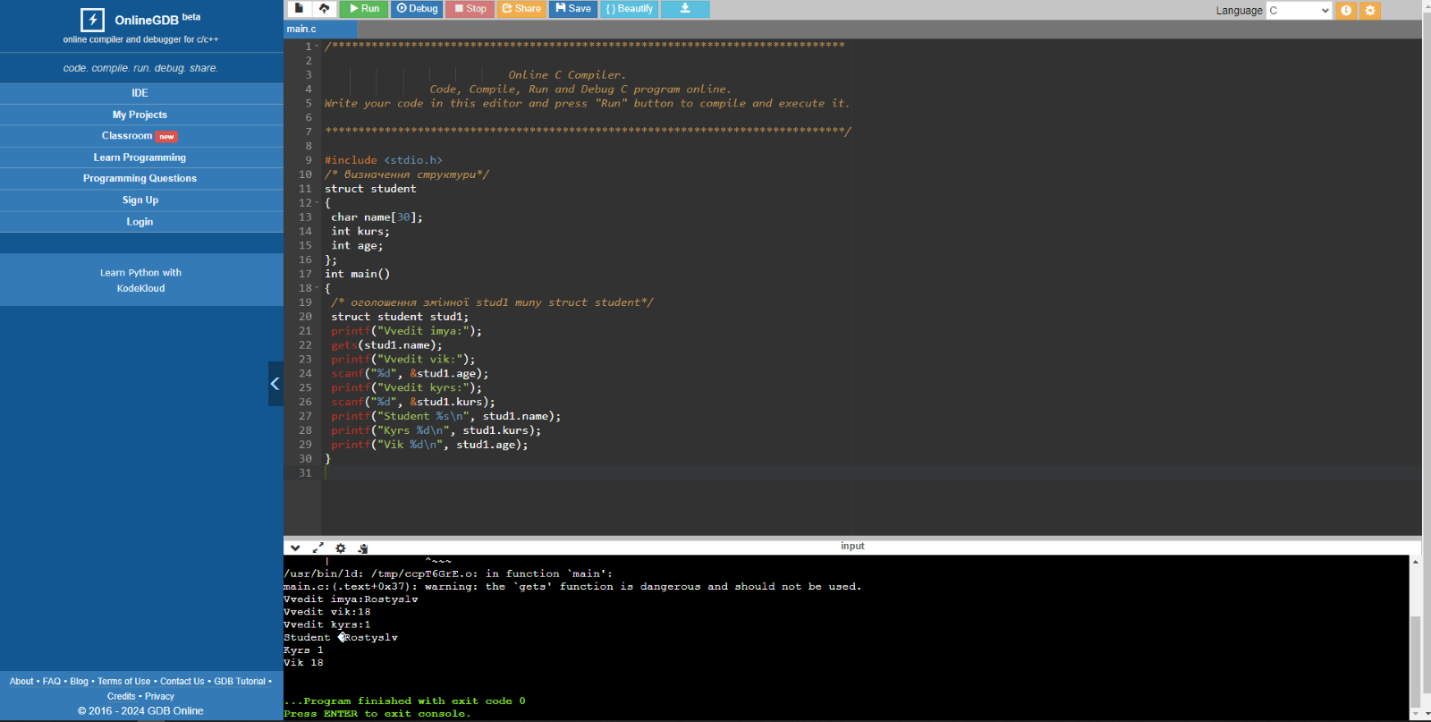
1. Ознайомитися з теоретичними відомостями.

2. Здійснити виконання прикладів, представлених у теоретичних відомостях, після чого представити скріни їх коду та результати виконання у звіті.

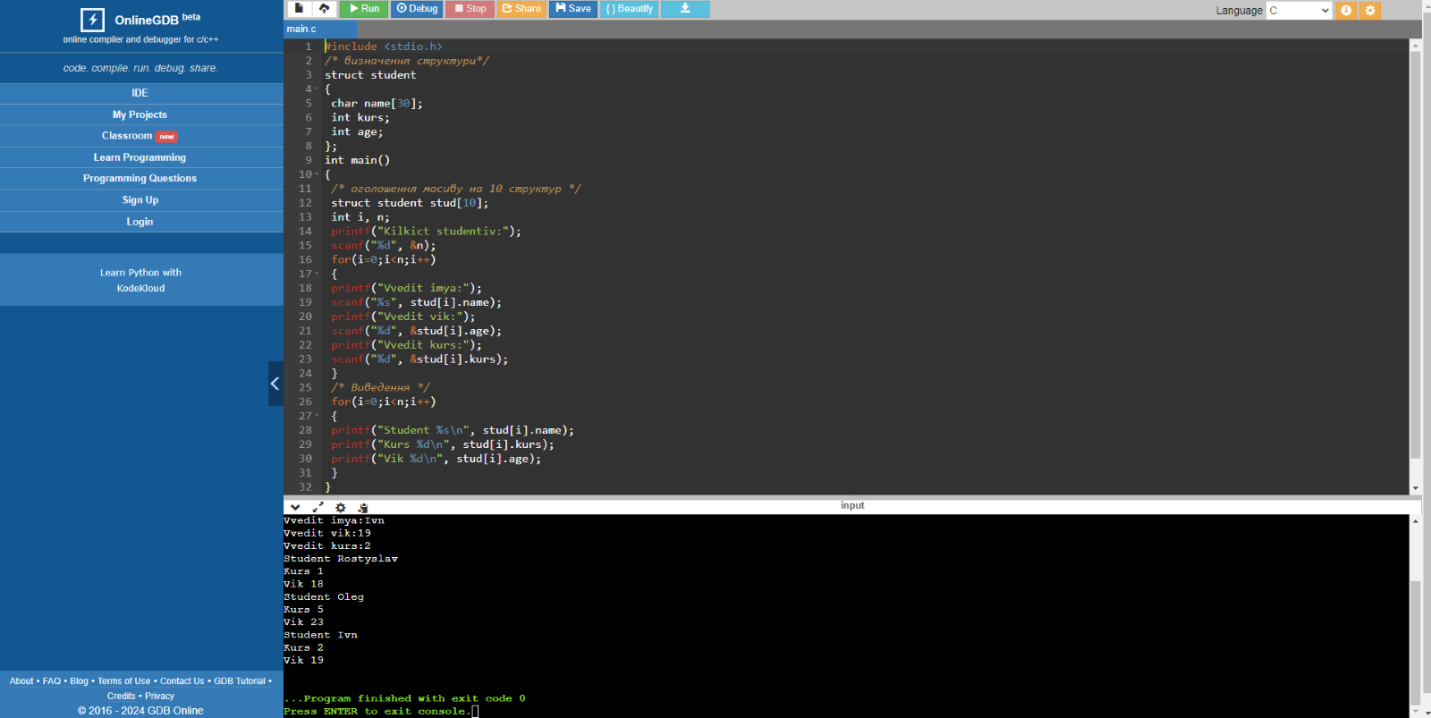
Присвоювання структур



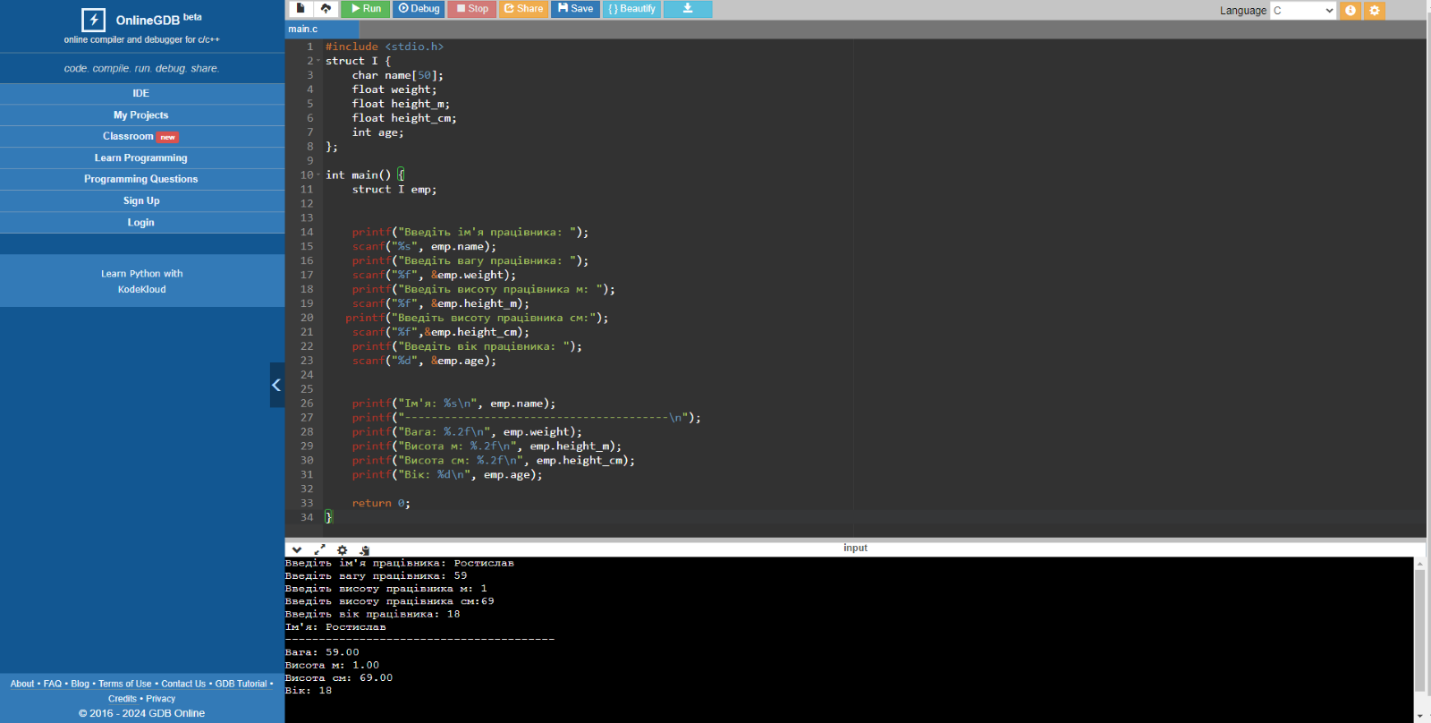
Програма для виведення певних видів даних про студента шляхом використання структури.



Програма, яка демонструє використання масиву структур.



3. Написати програму для виведення нижчепредставленої інформації шляхом використання структури. Ім’я, вага, висота, вік – вводяться з клавіатури ( вказати довільні дані).



Контрольні запитання:

1. Дайте визначення поняття «структура».

Структура – це сукупність змінних, об'єднаних під одним ім'ям. За допомогою структур зручно розміщувати в суміжних полях пов'язані між собою елементи інформації.

2. Яким чином здійснюється оголошення структури?

struct iм'я \_ структурного\_типу { тип\_поля iм'я\_поля ;

3. Охарактеризуйте синтаксис об’єднання даних.

Це спеціальний тип даних, який дозволяє зберігати різні типи даних у пам'яті, але спільною з ними пам'яті. Об'єднання дозволяє звертатися до різних полів за допомогою тієї самої пам'яті, тобто всі поля об'єднання починаються з тієї ж адреси.

4. Які операції не можна застосовувати до структур?

Неможливо використовувати оператори порівняння (==, !=, <, >, <=, >=) для структур.

struct Point { int x; int y; };

struct Point p1 = {1, 2}, p2 = {1, 2};

if (p1 == p2) { // Це некомпілюється

printf("p1 дорівнює p2");

}

Якщо структура містить масиви, присвоєння одного масиву іншому неможливе.

struct Data { int arr[10]; };

struct Data d1, d2;

d1 = d2; // Це компілюється, але дійсно копіює вміст масивів

Неможливо використовувати стандартні оператори вводу/виводу (наприклад, scanf, printf) безпосередньо для всієї структури. Введення та виведення треба виконувати поелементно.

struct Point { int x; int y; };

struct Point p = {1, 2};

printf("%d %d\n", p.x, p.y); // Виведення потрібно робити для кожного поля окремо