Міністерство освіти і науки України
Національний університет водного господарства та природокористування

Навчально-науковий інститут кібернетики, інформаційних технологій та інженерії
Кафедра обчислювальної техніки

04-04-280М

МЕТОДИЧНІ ВКАЗІВКИ

до лабораторних робіт
з навчальної дисципліни «**Програмування**»
для здобувачів вищої освіти першого (бакалаврського) рівня
за освітньо-професійною програмою «Комп'ютерна інженерія»
спеціальності 123 «Комп'ютерна інженерія»
та освітньо-професійною програмою «Інформаційна безпека»
спеціальності 125 «Кібербезпека та захист інформації»
денної та заочної форми навчання

Рекомендовано
науково-методичною радою
з якості ННІКІТІ
Протокол № 2 від 02.12.2024 р.

Рівне – 2024

Методичні вказівки до лабораторних робіт з навчальної дисципліни «Програмування» для здобувачів вищої освіти першого (бакалаврського) рівня за освітньо-професійною програмою «Комп'ютерна інженерія» спеціальності 123 «Комп'ютерна інженерія» та освітньо-професійною програмою «Інформаційна безпека» спеціальності 125 «Кібербезпека та захист інформації» денної та заочної форми навчання. [Електронне видання] / Сидор А. І., Бойчура М. В. – Рівне : НУВГП, 2024. – 47 с.

Укладачі: Сидор А. І., к.т.н., в.о. завідувача кафедри обчислювальної техніки;
Бойчура М. В., к.т.н., доцент кафедри обчислювальної техніки.

Відповідальний за випуск: Сидор А. І., к.т.н., в.о. завідувача кафедри обчислювальної техніки.

Керівник групи забезпечення
спеціальності
123 «Комп'ютерна інженерія» Сидор А. І.

Керівник групи забезпечення
спеціальності
125 «Кібербезпека та захист інформації» Назарук В. Д.

ЗМІСТ

Вступ

Методичні рекомендації до виконання лабораторних робіт з навчальної дисципліни Програмування створені для надання студентам базових знань та практичних навичок у програмуванні на мові C++. У цій частині студенти мають змогу ознайомитись з ключовими темами, які є основою для подальшого вивчення програмування. Кожна лабораторна робота спрямована на розвиток навичок роботи з основними конструкціями мови C++, такими як введення та виведення даних, математичні операції, використання функцій, а також правильне форматування коду.

У ході виконання лабораторних робіт студенти навчаться розробляти структуровані програми, ефективно використовувати цикли та функції, розуміти принципи форматування коду для полегшення його читабельності та підтримки. Кожна тема супроводжується теоретичними поясненнями, прикладами та завданнями для самостійного виконання, що сприяє кращому засвоєнню матеріалу.

Нижче наведено короткий огляд змісту лабораторних робіт.

Лабораторна робота №1 "Ввід та вивід інформації". Метою першої лабораторної роботи є ознайомлення студентів із основними конструкціями для введення та виведення даних. Студенти матимуть змогу навчитись працювати з різними типами даних і використовувати оператори введення/виведення для взаємодії з користувачем.

Лабораторна робота №2 "Використання математичних операцій". У другій лабораторній роботі розглядаються базові математичні операції. Студенти навчаться використовувати арифметичні оператори для

виконання обчислень та застосовувати математичні функції з бібліотеки <cmath> для розв'язання типових задач.

Лабораторна робота №3 "Використання функцій (C++)". Третя лабораторна робота присвячена створенню та використанню функцій у програмуванні. Вона допоможе зрозуміти, як функції допомагають структурувати код і робити його більш зручним для читання та підтримки. Студенти навчяться передавати параметри у функції та отримувати результати їх виконання.

Лабораторна робота №4 "Базове форматування коду". У четвертій лабораторній роботі студенти навчаться правильно форматувати код, дотримуючись стандартних правил. Це надасть розуміння важливості правильного розташування відступів, використання логічних імен змінних, коментарів та порожніх рядків для полегшення роботи з кодом.

Лабораторна робота №5 "Робота з декількома функціями та використання циклу for". П'ята лабораторна робота допоможе освоїти складніші аспекти програмування, зокрема роботу з декількома функціями та використання циклів для виконання повторюваних дій. Студенти навчаться використовувати цикл for для обробки наборів даних і виконання завдань, що потребують ітерацій.

Виконання цих лабораторних робіт дозволить студентам закріпити основи програмування та підготуватися до більш складних тем, які будуть розглянуті в наступних частинах курсу.

Лабораторна робота №1

Тема: Ввід та вивід інформації

1. Мета лабораторної роботи

Ознайомити студентів з основними операторами введення та виведення інформації в мові програмування C++. Навчити працювати з базовими типами даних та виконувати основні операції введення/виведення.

2. Навчальні результати

У результаті виконання лабораторної роботи студент зможе:

- Використовувати оператори cin та cout для введення/виведення інформації в C++.
- Працювати з базовими типами даних: цілі числа, числа з плаваючою комою, символи, рядки.
- Виконувати прості арифметичні операції над введеними даними та виводити результат на екран.

3. Теоретичні відомості

У C++ основні операції введення та виведення виконуються за допомогою бібліотеки iostream.

Оператор введення cin: використовується для введення даних з клавіатури.

Оператор виведення cout: використовується для виведення даних на екран.

Маніпулятори endl і \n: використовуються для переходу на новий рядок у виведенні.

Синтаксис:

```
    int x;
    cin >> x;  // Введення числа з клавіатури
    cout << "Введене число: " << x << endl;    // Виведення числа
```

Приклад програми:

```
#include <iostream>
#include <iomanip>
using namespace std;

int main() {
   string name;
   int age;

   // Введення імені та віку
   cout << "Введіть ваше ім'я: ";
   cin >> name;
   cout << "Введіть ваш вік: ";
   cin >> age;

   // Виведення імені та віку
   cout << "Ваше ім'я: " << name << endl;
   cout << "Ваш вік: " << age << endl;

   // Арифметична операція
   int a, b;
   cout << "Введіть два числа для додавання: ";
   cin >> a >> b;
   cout << "Сума: " << a + b << endl;

   // Введення числа з плаваючою комою та форматований вивід
   double number;
   cout << "Введіть число з плаваючою комою: ";
```

```
    cin >> number;
    cout << "Число з точністю до двох знаків після коми: "
        << fixed << setprecision(2) << number << endl;

    return 0;
}
```

4. Допоміжні конструкції

iostream – бібліотека і відповідний заголовний файл, який використовується для організації введення-виведення в мові програмування С++.

#include <ім'я_файлу>. Директива #include використовується для включення копії вказаного файлу в те місце програми, де знаходиться ця директива.

using namespace std; повідомляє компілятору, що ми хочемо використовувати все, що знаходиться в просторі імен std.

"int main()" – це вираз, що показує, що в програмі присутня головна функція main(), яка поверне як значення ціле число.

std::cout використовується для виведення значення (console out = вивід); std::cin використовується для отримання значення (console in = ввід).

cout << "Текст";

cin >> змінна ;

Цикл while служить для організації N-кратного виконання групи операторів (тіла циклу) до тих пір, поки залишається істинною умова виконання циклу:

while (умова) {}

|| – оператор логічного «або».

&& – оператор логічного «І».

int – це тип даних, який використовується для представлення реальних чисел, що не мають дробових значень:

int змінна; – інструкція для оголошення змінної.

cin.ignore(32767, '\n'); //видаляє значення попереднього введення із вхідного буфера.

cin.clear(); // повернення cin в «звичайний» режим роботи.

\n (newline), endl (end of line) – перенесення каретки на наступний рядок.

if (умова) – умовний оператор, який набуває істинного значення, якщо виконується умова в дужках.

return 0; // повернення числа 0, оскільки основна функція створює числове значення для int main(). Термін return використовується для повернення результату за допомогою функції. Це означає, що програма була точно виконана, і ми можемо використовувати оператор return для завершення основної функції. Коли основна функція має тип даних «ціло число», вона повинна щось повертати.

5. Завдання

Завдання: №1

Напишіть програму, яка запитує у користувача номер місяця і потім виводить відповідну назву пори року. У разі, якщо користувач ввів неприпустиме число, програма повинна вивести повідомлення про помилку.

Алгоритм програми: підключаємо бібліотеку-викликаємо простір імен std-запускаємо головну функцію-виводимо на екран повідомлення – оголошуємо змінну-записуємо ведене число в зміну – перевіряємо правильність веденого числа – виконуємо ведення поки не введено вірне число – повертаємо cin в звичайний режим – виводимо пору року відповідно до місяця.

Завдання №2

Напишіть програму обчислення вартості покупки з урахуванням знижки. Знижка в 3% надається, якщо сума

покупки складає менше 1000 гривень, в 5% – якщо сума покупки складає більше 1000 гривень.

Завдання №3

Виконайте одне додаткове завдання згідно варіанту, співставивши Ваш номер у журналі та номер варіанту у таблиці.

Завдання згідно варіанту

№ варіанту	Завдання
1	**Модифікація завдання 1:** Додати виведення повідомлення про кількість днів у введеному місяці.
2	**Модифікація завдання 1:** Визначити пору року та день тижня для першого числа введеного місяця.
3	**Модифікація завдання 1:** Запитати у користувача додатково рік та врахувати високосні роки для лютого (29 днів).
4	**Модифікація завдання 2:** Якщо сума покупки перевищує 5000 гривень, надати додаткову знижку 10%.
5	**Модифікація завдання 2:** Додати можливість вибору валюти (гривня, долар, євро) та перерахувати ціну за поточним курсом.
6	**Модифікація завдання 2:** Додати можливість отримати додаткову знижку у розмірі 7%, якщо клієнт є постійним покупцем.
7	**Модифікація завдання 2:** Додати систему накопичувальних знижок: 3% – для покупок до 1000 грн, 5% – для покупок до 5000 грн, 7% – для покупок понад 5000 грн.
8	**Модифікація завдання 2:** Якщо покупка складає рівно

	1000 грн, надати вибір знижки між 3% і 5%.
9	**Завдання:** Написати програму, яка запитує у користувача кількість покупок і обчислює загальну суму зі знижками для кожної покупки.
10	**Завдання:** Написати програму для перерахунку загальної суми покупок у різних валютах (гривня, долар, євро).
11	**Завдання:** Написати програму, яка запитує у користувача день та місяць і визначає, чи є цей день святковим.
12	**Завдання:** Написати програму для визначення днів у місяці залежно від пори року.
13	**Завдання:** Написати програму для обчислення знижок для різних категорій покупців (звичайний, студент, пенсіонер).
14	**Завдання:** Модифікувати завдання 1, додавши перевірку на існування такого місяця на іншій планеті (уявне завдання).
15	**Завдання:** Написати програму, яка запитує у користувача кілька місяців і виводить пори року для кожного.
16	**Завдання:** Написати програму для обчислення вартості кількох товарів з різними знижками.
17	**Завдання:** Написати програму для виведення кількості днів до початку найближчої пори року, залежно від поточного місяця.
18	**Завдання:** Написати програму, яка запитує у користувача номер дня року і виводить відповідний місяць та день.

19	**Завдання:** Написати програму, яка обчислює середню знижку при купівлі кількох товарів.
20	**Завдання:** Модифікувати завдання 2, додавши функцію порівняння знижок для кількох магазинів.
21	**Завдання:** Написати програму для обчислення суми покупок з урахуванням податку на додану вартість (ПДВ).
22	**Завдання:** Написати програму для визначення місяця за введеним номером кварталу (1-4).
23	**Завдання:** Написати програму, яка обчислює кількість місяців до найближчої весни/зими/літа/осені.
24	**Завдання:** Модифікувати завдання 2, додавши опцію накопичення балів за покупку з можливістю їх використання як знижки на наступну покупку.
25	**Завдання:** Написати програму, яка запитує номер місяця і визначає, у скільки разів вартість покупки підвищується в різні пори року.
26	**Завдання:** Написати програму, яка обчислює загальну вартість покупки з різними знижками на різні види товарів (електроніка, одяг, продукти).
27	**Завдання:** Написати програму для обчислення вартості покупки з урахуванням сезонних знижок (наприклад, на зимові товари влітку).
28	**Завдання:** Написати програму для розрахунку вартості покупки з додатковою послугою доставки та її знижкою залежно від вартості товару.
29	**Завдання:** Написати програму для розрахунку вартості покупки з можливістю вибору способу оплати (готівка/карта) та різними знижками для кожного

	варіанту.
30	**Завдання:** Модифікувати завдання 2, додавши опцію вибору покупцем кількості товарів, і надавати знижку залежно від кількості одиниць товару (чим більше товарів – тим більша знижка).

6. Рекомендації щодо усунення помилок

- Помилки введення. Переконайтеся, що тип змінної відповідає типу введених даних. Якщо Ви намагаєтеся ввести рядок замість числа, програма виведе помилку.
- Формат виведення. Для точного виведення чисел використовуйте маніпулятори fixed та setprecision.

Додаткові матеріали

Для набуття практичних навичок використовувати матеріали за наступними посиланнями:

1. https://www.geeksforgeeks.org/basic-input-output-c/
2. https://cplusplus.com/doc/tutorial/basic_io/
3. https://www.programiz.com/cpp-programming/input-output

Лабораторна робота №2

Тема: Використання математичних операцій

1. Мета лабораторної роботи

Навчити студентів використовувати базові математичні операції в мові програмування C++. Засвоїти операції додавання, віднімання, множення, ділення, а також обчислення залишку від ділення. Навчитись використовувати математичні функції зі стандартної бібліотеки C++.

2. Навчальні результати

Після виконання лабораторної роботи студент зможе:

- Використовувати базові арифметичні оператори в C++.
- Застосовувати математичні функції з бібліотеки <cmath>.
- Писати програми для обчислення арифметичних виразів і вирішення математичних задач.

3. Теоретичні відомості

У C++ доступні такі базові арифметичні операції:

Додавання (+): a + b

Віднімання (-): a - b

Множення (*): a * b

Ділення (/): a / b

Остача від ділення (%): a % b (тільки для цілих чисел)

Математичні функції з бібліотеки <cmath>:

sqrt(x) – обчислення квадратного кореня.

pow(x, y) – піднесення числа x до степеня y.
abs(x) – модуль числа x.
sin(x), cos(x), tan(x) – тригонометричні функції.
log(x) – натуральний логарифм числа x.

Синтаксис підключення бібліотеки:

```
#include <cmath>
```

Приклад програми:

```
#include <iostream>
#include <cmath>
using namespace std;

int main() {
    double a, b;

    // Введення двох чисел
    cout << "Введіть два числа: ";
    cin >> a >> b;

    // Арифметичні операції
    cout << "Сума: " << a + b << endl;
    cout << "Різниця: " << a - b << endl;
    cout << "Добуток: " << a * b << endl;
    cout << "Частка: " << a / b << endl;

    // Остача від ділення (для цілих чисел)
    int intA = static_cast<int>(a), intB = static_cast<int>(b);
    cout << "Остача від ділення: " << intA % intB << endl;

    return 0;
}
```

4. Допоміжні конструкції

Для підключення кирилиці підключіть бібліотеку #include <Windows.h> та в головній функції введіть SetConsoleCP(1251); SetConsoleOutputCP(1251);

5. Завдання

Завдання №1

Напишіть програму обчислення вартості поїздки на автомобілі на дачу (туди і назад). Вихідними даними є:

- відстань до дачі (в км);
- літраж бензину, який споживає автомобіль на 100 км пробігу;
- ціна одного літра бензину.

Приклад результату виконання програми:

Відстань до дачі (км): 67
Витрати бензину (літрів на 100 км пробігу): 8.5
Ціна літра бензину (грн.): 25
Поїздка на дачу і назад обійдеться в 284.75 грн.

Завдання №2

Напишіть програму, яка порівнює два введених з клавіатури числа. Програма повинна вказати, яке число менше, або, якщо числа рівні, – вивести відповідне повідомлення.

Приклад результату виконання програми:

Введіть 2 цілих числа: 48 54

Приклад результату виконання програми:

48 менше 54

Завдання №3

Виконайте одне додаткове завдання згідно варіанту, співставивши Ваш номер у журналі та номер варіанту у таблиці.

Завдання згідно варіанту

№ варіанту	Завдання
1	**Модифікація завдання 1:** Додати можливість враховувати витрати на додаткові послуги (наприклад, паркування або платні дороги).
2	**Модифікація завдання 1:** Додати введення часу поїздки та обчислити середню швидкість автомобіля.
3	**Модифікація завдання 1:** Додати обчислення вартості поїздки з урахуванням змінної ціни на бензин (наприклад, різні ціни в різних регіонах).
4	**Модифікація завдання 1:** Додати можливість обчислити вартість поїздки для різних транспортних засобів (автомобіль, автобус, електромобіль).
5	**Модифікація завдання 1:** Обчислити загальний час поїздки, враховуючи середню швидкість автомобіля.
6	**Модифікація завдання 2:** Додати перевірку, чи є числа цілими або дробовими.
7	**Модифікація завдання 2:** Якщо введені числа рівні, надати користувачу можливість ввести ще одне число для порівняння.
8	**Завдання:** Написати програму для обчислення відстані, яку можна проїхати за дану суму грошей.
9	**Завдання:** Обчислити кількість поїздок на дачу, яку можна зробити з фіксованою кількістю грошей.
10	**Завдання:** Додати можливість вказати тип палива і обчислити вартість поїздки для бензину, дизелю і газу.
11	**Завдання:** Написати програму для порівняння витрат на поїздку в різні сезони, враховуючи різні ціни на паливо.

12	**Завдання:** Написати програму, яка обчислює вартість поїздки залежно від кількості пасажирів (розділення вартості між усіма пасажирами).
13	**Завдання:** Модифікувати завдання 2, додавши можливість порівняти три числа.
14	**Завдання:** Додати до завдання 2 обчислення модуля різниці між двома числами.
15	**Завдання:** Написати програму, яка обчислює загальну кількість пального, необхідну для кількох поїздок на дачу за місяць.
16	**Завдання:** Написати програму для визначення найбільш економічного маршруту на дачу (кілька маршрутів з різними відстанями і цінами на паливо).
17	**Завдання:** Написати програму, яка порівнює кілька автомобілів за ефективністю використання пального.
18	**Завдання:** Написати програму для обчислення вартості поїздки, враховуючи пробки (збільшення витрат на паливо через затримки).
19	**Завдання:** Написати програму для обчислення вартості поїздки з урахуванням знижок на паливо (бонусні програми АЗС).
20	**Завдання:** Модифікувати завдання 2, додавши можливість порівняти числа за абсолютною величиною (модуль чисел).
21	**Завдання:** Написати програму для порівняння витрат на поїздку в різні країни (різні ціни на паливо і вартість дороги).
22	**Завдання:** Написати програму для розрахунку витрат на поїздку для гібридного автомобіля, враховуючи

	використання електрики і бензину.
23	**Завдання:** Написати програму, яка обчислює загальну вартість поїздок на дачу за рік.
24	**Завдання:** Написати програму для обчислення залишкової кількості пального після поїздки (враховуючи початкову кількість в баку).
25	**Завдання:** Додати можливість обчислювати витрати на подорож не тільки на автомобілі, але і на інших видах транспорту (авіа, залізничний транспорт).
26	**Завдання:** Написати програму для визначення найдешевшого варіанту палива, враховуючи різні ціни на різних АЗС.
27	**Завдання:** Додати можливість обчислювати витрати на поїздку з урахуванням різних умов (їзда в горах, погані дороги тощо).
28	**Завдання:** Написати програму, яка порівнює кілька поїздок за різними маршрутами і вибирає найдешевший варіант.
29	**Завдання:** Модифікувати завдання 2, додавши можливість виводити на екран проміжні кроки порівняння (якщо числа однакові, запитувати наступне число).
30	**Завдання:** Додати можливість обчислювати витрати на поїздку з урахуванням відвідування декількох точок на шляху (наприклад, кілька дач або зупинок).

6. Рекомендації щодо усунення помилок

- Якщо програма виводить неправильні результати, перевірте правильність введення

чисел та використання відповідних операторів або функцій.

- Якщо ділення на нуль викликає помилку, додайте перевірку на нульове значення перед операцією ділення.

Додаткові матеріали

Для набуття практичних навичок використовувати матеріали за наступними посиланнями:

1. https://www.w3schools.com/cpp/cpp_operators.asp
2. https://www.geeksforgeeks.org/cpp-arithmetic-operators/
3. https://www.programiz.com/cpp-programming/operators

Лабораторна робота №3

Тема: Використання функцій (C++)

1. Мета лабораторної роботи

Ознайомити студентів з принципами створення та використання функцій у програмуванні на мові C++. Навчити передавати параметри у функції, використовувати функції для вирішення типових завдань і повертати результати виконання функцій.

2. Навчальні результати

Після виконання лабораторної роботи студент зможе:

- Створювати та використовувати функції у програмах.
- Передавати параметри у функції за значенням та за посиланням.
- Повертати результати функцій.
- Писати програми з розбиттям на логічні частини, використовуючи функції для повторюваних дій.

3. Теоретичні відомості

У C++ функція – це блок коду, який можна викликати для виконання певної задачі. Вона може приймати параметри і повертати результат. Створення функцій допомагає уникати дублювання коду та робить програму більш структурованою.

Загальний синтаксис функції:

```
return_type function_name(parameter_list) {
    // Тіло функції
    return value;  // якщо функція повертає результат
}
```

Роль функцій.

Функції допомагають зменшити рівень дублювання коду. Якщо функціональність виконується в кількох місцях програмного забезпечення, тоді замість того, щоб писати той самий код знову і знову, ми створюємо функцію та викликаємо її всюди. Це також допомагає в обслуговуванні, оскільки нам доведеться вносити зміни лише в одному місці, якщо ми вносимо зміни до функціональності в майбутньому.

Функції роблять код модульним. Розглянемо великий файл, що містить багато рядків коду. Читати та використовувати код стає дуже просто, якщо код розділити на функції.

Функції забезпечують абстракцію. Наприклад, ми можемо використовувати бібліотечні функції, не турбуючись про їх внутрішню роботу.

Функція C++ складається з двох частин:

- Оголошення: тип повернення, назва функції та параметри (якщо є)
- Визначення: тіло функції (код для виконання)

Приклад програми:

```
#include <iostream>
using namespace std;
```

```
int sum(int a, int b) {
   return a + b;
}

int main() {
   int x, y;
   cout << "Введіть два числа: ";
   cin >> x >> y;

   cout << "Сума: " << sum(x, y) << endl;

   return 0;
}
```

4. Допоміжні конструкції

Більшість програм C++ мають функцію під назвою main(), яка викликається операційною системою, коли користувач запускає програму.

Кожна функція має тип повернення. Якщо функція не повертає жодного значення, тоді як тип повернення використовується void. Більше того, якщо тип повернення функції недійсний, ми все ще можемо використовувати оператор return у тілі визначення функції, не вказуючи жодної константи, змінної тощо разом із ним, лише згадавши «return;».

5. Завдання

Завдання №1

Напишіть програму, яка обчислює суму перших n цілих додатних чисел. Кількість чисел вводить користувач. Програму реалізувати через три функції (ввід числа, обрахунок суми, вивід числа).

Приклад результату виконання програми:

Введіть кількість чисел: 9

Сума перших 9 цілих додатних чисел дорівнює 45

Завдання №2

Напишіть програму для переведення ваги з фунтів в кілограми (1 фунт = 0.453 кг). Реалізувати це через функції.

Приклад результату виконання програми:

Введіть вагу в фунтах: 12.4

12.4 фунтів = 5.6172 кг

Завдання №3

Виконайте одне додаткове завдання згідно варіанту, співставивши Ваш номер у журналі та номер варіанту у таблиці.

Завдання згідно варіанту

№ варіанту	Завдання
1	**Модифікація завдання 1:** Додайте перевірку на коректність введеного числа (негативні або дробові числа не допускаються).
2	**Модифікація завдання 1:** Реалізуйте функцію, яка обчислює суму тільки парних чисел з n.
3	**Модифікація завдання 1:** Додайте функцію для обчислення суми квадратів перших n чисел.
4	**Модифікація завдання 1:** Реалізуйте обчислення суми перших n непарних чисел.
5	**Модифікація завдання 1:** Додайте функцію для обчислення середнього арифметичного перших n чисел.
6	**Модифікація завдання 1:** Реалізуйте функцію для обчислення добутку перших n чисел.

7	**Модифікація завдання 1:** Додайте функцію для обчислення суми перших n чисел, що діляться на 3.
8	**Модифікація завдання 1:** Напишіть програму, яка обчислює суму перших n цілих чисел, кратних 5.
9	**Модифікація завдання 1:** Додайте функцію для обчислення суми перших n чисел за допомогою рекурсії.
10	**Модифікація завдання 1:** Додайте функцію для обчислення суми чисел, кратних якому-небудь числу, введеному користувачем.
11	**Модифікація завдання 2:** Додайте можливість користувачу конвертувати вагу також з кілограмів у фунти.
12	**Модифікація завдання 2:** Додайте можливість введення кількох значень ваги у фунтах для перерахунку в кілограми (через масив).
13	**Модифікація завдання 2:** Додайте функцію для перетворення ваги з фунтів в інші одиниці виміру (наприклад, грами, тонни).
14	**Модифікація завдання 2:** Додайте можливість користувачу вводити кілька ваг в фунтах і обчислювати їх середнє значення у кілограмах.
15	**Модифікація завдання 2:** Реалізуйте програму для перерахунку не тільки ваги, але й відстані (наприклад, з миль у кілометри).
16	**Модифікація завдання 2:** Додайте функцію для автоматичного округлення результатів до двох знаків після коми.
17	**Модифікація завдання 2:** Додайте можливість перетворення ваги з інших одиниць виміру, таких як

	унції, грами і тонни.
18	**Завдання:** Напишіть програму для обчислення суми чисел, кратних 7, з використанням функцій.
19	**Завдання:** Напишіть програму для знаходження найменшого спільного кратного (НСК) двох чисел за допомогою функцій.
20	**Завдання:** Напишіть програму для перетворення часу (з секунд у хвилини, години та дні) з використанням функцій.
21	**Завдання:** Напишіть програму для перетворення температури з градусів Фаренгейта в Цельсій за допомогою функцій.
22	**Завдання:** Напишіть програму для знаходження найбільшого спільного дільника (НСД) двох чисел за допомогою функцій.
23	**Завдання:** Реалізуйте програму, яка знаходить корені квадратного рівняння за допомогою функцій.
24	**Завдання:** Напишіть програму для обчислення об'єму циліндра за допомогою функцій.
25	**Завдання:** Додайте функцію для перетворення площі з квадратних метрів у квадратні ярди.
26	**Завдання:** Напишіть програму для обчислення вартості поїздки з урахуванням витрати палива, використовуючи функції.
27	**Завдання:** Реалізуйте програму для конвертації валют (долари, євро, гривні) з використанням функцій.
28	**Завдання:** Напишіть програму для обчислення площі трикутника за трьома сторонами (формула Герона) через

	функції.
29	**Завдання:** Напишіть програму для перетворення числа у різні системи числення (двійкова, вісімкова, шістнадцяткова) через функції.
30	**Завдання:** Реалізуйте програму для обчислення середнього балу за декількома оцінками з використанням функцій.

6. Рекомендації щодо усунення помилок:
   - Переконайтеся, що кожна функція виконує одну конкретну задачу. Не перевантажуйте функції зайвою логікою.
   - Якщо функція повертає некоректний результат, перевірте правильність використання операторів і умови у функції.

Додаткові матеріали

Для набуття практичних навичок використовувати матеріали за наступним посиланням:

https://www.w3schools.com/cpp/cpp_functions.asp

Лабораторна робота №4

Тема: Базове форматування коду

1. Мета лабораторної роботи

Ознайомити студентів з основними принципами базового форматування коду в мові програмування C++. Навчити писати код, який буде легко читати, розуміти та підтримувати. Важливим є дотримання правил відступів, правильне використання порожніх рядків, коментарів та імен змінних.

2. Навчальні результати

Після виконання лабораторної роботи студент зможе:

- Використовувати відступи для покращення читабельності коду.
- Правильно називати змінні, функції та інші елементи коду.
- Використовувати коментарі для пояснення логіки роботи програми.
- Дотримуватися базових правил структурованого кодування.

3. Теоретичні відомості

Основні правила базового форматування коду в C++:

1. Відступи та структура коду:

- Стандартний відступ – 4 пробіли або один символ табуляції.

- Всі блоки коду (наприклад, в тілі циклів, умовних операторів, функцій) повинні мати відступи.
- Відкриваюча дужка { розташовується на тій самій лінії, що й оголошення конструкції (функція, цикл тощо).
- Закриваюча дужка } повинна бути на новій лінії під початковим блоком.

Приклад:

```
if (x > 0) {
   cout << "Додатне число" << endl;
} else {
   cout << "Від’ємне число" << endl;
}
```

2. Назви змінних та функцій:

- Використовуйте зрозумілі імена для змінних і функцій.
- Для імен змінних застосовуйте стиль camelCase або snake_case (наприклад, userName або user_name).
- Імена функцій мають бути дієсловами, що описують дію (наприклад, calculateSum(), getUserInput()).

3. Коментарі:

- Використовуйте коментарі для пояснення складних або важливих частин коду.
- Коментарі можуть бути однорядковими (//) або багаторядковими (/* ... */).

Приклад:
```
// Це функція для обчислення суми двох чисел
int sum(int a, int b) {
   return a + b;
}
```

4. Порожні рядки:

- Використовуйте порожні рядки для логічного розбиття коду на блоки.
- Наприклад, додайте порожні рядки перед і після визначення функцій, щоб код виглядав структуровано.

5. Константи:

- Якщо значення не буде змінюватись у процесі виконання програми, використовуйте ключове слово const для оголошення констант.

Приклад:
```
const double PI = 3.14159;
```

Приклад програми:
```
#include <iostream>
using namespace std;

// Функція для перевірки парності числа
bool isEven(int number) {
   return number % 2 == 0;
}

int main() {
```

```
    int num;

    // Введення числа від користувача
    cout << "Введіть число: ";
    cin >> num;

    // Перевірка парності
    if (isEven(num)) {
        cout << "Число парне." << endl;
    } else {
        cout << "Число непарне." << endl;
    }

    return 0;
}
```

4. Допоміжні конструкції:

Цикл While у C++ використовується в ситуаціях, коли ми заздалегідь не знаємо точну кількість ітерацій циклу. Виконання циклу припиняється на основі тестової умови. Цикли в C++ починають використовуватися, коли нам потрібно багаторазово виконувати блок операторів.

Синтаксис:

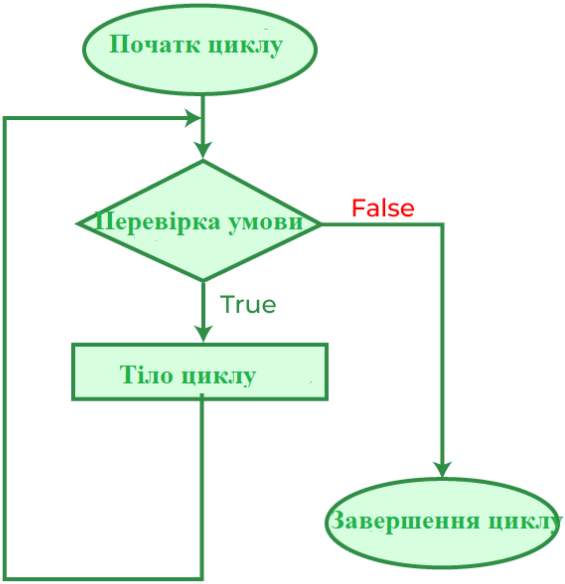
```
while (тестовий_вираз)
{
  // тіло

  оновлення_виразу;
}
```

Різні частини циклу While:

- Тестовий вираз: у цьому виразі ми маємо перевірити умову. Якщо умова оцінюється як

істина, тоді ми виконаємо тіло циклу та перейдемо до виразу оновлення. В іншому випадку ми вийдемо з циклу while.

- Оновлення виразу: після виконання тіла циклу цей вираз збільшує/зменшує змінну циклу на деяке значення.
- Тіло: це група операторів, які містять змінні, функції тощо. За допомогою циклу while можна друкувати код і прості імена, виконувати складні алгоритми або функціональні операції.



Цикл While виконується наступним чином:

1. Керування надається циклу while.

2. Потік переходить до стану перевірки Умови.
3. Якщо Умова виконується, то потік надходить у Тіло. В інакшому випадку потік покидає межі циклу.
4. Оператори всередині тіла циклу виконуються.
5. Керування знову повертається до кроку 2.

5. Завдання

Завдання №1

Напишіть програму для перерахунку величини часового інтервалу, заданого в хвилинах, в величину, виражену в годинах і хвилинах (реалізувати через дві функції: ввід числа та обрахунок).

Приклад результату виконання програми:

Введіть часовий інтервал (в хвилинах): 150
150 хвилин = 2 год. 30 хв.

Завдання №2

Напишіть програму, яка виводить на екран таблицю вартості, наприклад, яблук в діапазоні від 100 г до 1 кг з кроком в 100 г.

Приклад результату виконання програми:

Введіть ціну за 1 кг яблук: 16.50

Вага (г)	Вартість (грн.)
100	1.65
200	3.30
300	4.95
400	6.60
500	8.25
600	9.90
700	11.55
800	13.20

900 14.85
1000 16.50

Завдання №3

Виконайте одне додаткове завдання згідно варіанту, співставивши Ваш номер у журналі та номер варіанту у таблиці.

Завдання згідно варіанту

№ варіанту	Завдання
1	**Модифікація завдання 1:** Додати можливість вводити кількість хвилин відразу для кількох значень і обчислювати результати для кожного.
2	**Модифікація завдання 1:** Реалізувати функцію для перетворення годин і хвилин назад у хвилини.
3	**Модифікація завдання 1:** Додати можливість користувачеві вводити час у годинах і хвилинах та отримувати загальну кількість хвилин.
4	**Модифікація завдання 1:** Додати перевірку на негативні значення хвилин.
5	**Модифікація завдання 1:** Реалізувати програму, яка буде виводити різницю між двома часовими інтервалами, заданими у хвилинах.
6	**Модифікація завдання 1:** Додати функцію, яка перетворює хвилини в дні, години і хвилини (наприклад, 1440 хвилин = 1 день).
7	**Модифікація завдання 1:** Додати можливість вводити інтервал часу в секундах і перетворювати їх у години та хвилини.
8	**Модифікація завдання 1:** Реалізувати програму для обчислення часу, що залишився до кінця дня, якщо дано

	поточний час у хвилинах.
9	**Модифікація завдання 2:** Додати можливість вводити вагу яблук в будь-якому діапазоні, наприклад, від 50 г до 2 кг з довільним кроком.
10	**Модифікація завдання 2:** Додати можливість виводити таблицю вартості для кількох фруктів (яблука, груші, апельсини).
11	**Модифікація завдання 2:** Реалізувати програму для обчислення вартості не тільки яблук, але й інших товарів з різними кроками (наприклад, 200 г для бананів).
12	**Модифікація завдання 2:** Додати можливість виводити вартість не тільки за грамами, але й за кількістю товару (наприклад, за 1, 2, 3 штуки).
13	**Модифікація завдання 2:** Додати можливість виводити таблицю вартості для декількох товарів одночасно, використовуючи масиви для збереження цін.
14	**Модифікація завдання 2:** Додати можливість виводити таблицю вартості з урахуванням знижки (наприклад, 10% знижка).
15	**Модифікація завдання 2:** Реалізувати можливість введення ціни за товар в іншій валюті та автоматичний перерахунок у гривні.
16	**Модифікація завдання 2:** Додати функцію для обчислення вартості кількох різних ваг товару (наприклад, 150 г, 450 г).
17	**Модифікація завдання 2:** Реалізувати можливість виводити таблицю вартості в різних одиницях виміру (г, кг, тонни).
18	**Завдання:** Написати програму для виведення таблиці вартості бензину за різними цінами і кількістю літрів

	(від 1 до 10 літрів).
19	**Завдання:** Реалізувати програму, яка виводить таблицю вартості будь-якого товару, залежно від кількості грамів або штук з довільним кроком.
20	**Завдання:** Написати програму для обчислення вартості товару за двома різними тарифами (наприклад, ціна за 500 г і ціна за 1 кг).
21	**Завдання:** Додати функцію для виведення таблиці вартості товару залежно від ваги та додаткових параметрів (наприклад, за різної якості товару).
22	**Завдання:** Додати функцію для автоматичного підрахунку кількості упаковок товару (наприклад, яблук у коробках по 1 кг).
23	**Завдання:** Написати програму для автоматичного розрахунку вартості доставки залежно від ваги товару.
24	**Завдання:** Реалізувати програму, яка автоматично розраховує загальну вартість товару за різними цінами для різних покупців (оптових та роздрібних).
25	**Завдання:** Написати програму для обчислення вартості товару з урахуванням змінної ціни за кожен наступний кілограм (наприклад, знижка на більші ваги).
26	**Завдання:** Реалізувати програму для підрахунку загальної вартості товару за кількома різними критеріями (ціна за 1 кг, кількість, знижки).
27	**Завдання:** Додати функцію для порівняння вартості двох різних товарів за заданою вагою (наприклад, яблука і груші).
28	**Завдання:** Написати програму, яка обчислює вартість товару залежно від ваги та об'єму упаковки (наприклад,

	для рідин чи сипучих продуктів).
29	**Завдання:** Реалізувати програму для обчислення вартості різних товарів з урахуванням податку або додаткових витрат (наприклад, ПДВ).
30	**Завдання:** Додати функцію для автоматичного перерахунку ціни залежно від валюти, використовуючи поточний курс обміну.

6. Рекомендації щодо усунення помилок:
   - Якщо код важко читати, перевірте, чи дотримуєтеся Ви правильних відступів та вказуєте порожні рядки між різними частинами коду.
   - Переконайтеся, що всі змінні мають змістовні імена, які відображають їх призначення.
   - Якщо виникають складнощі з розумінням логіки програми, додайте більше коментарів.

Додаткові матеріали

Для набуття практичних навичок використовувати матеріали за наступними посиланнями:

1. https://google.github.io/styleguide/cppguide.html
2. https://codebeautify.org/cpp-formatter-beautifier
3. https://doc.qt.io/qtcreator/creator-preferences-cpp-code-style.html

Лабораторна робота №5

Тема: Робота з декількома функціями та використання циклу for

1. Мета лабораторної роботи

Набуття навичок використання декількох функцій у програмі, ознайомлення з організацією циклів, зокрема циклу for. Навчити передавати значення між функціями та ефективно використовувати цикл для повторення операцій.

2. Навчальні результати

Після виконання лабораторної роботи студент зможе:

- Створювати програми, що використовують декілька функцій для організації різних частин коду.
- Розуміти і застосовувати цикл for для виконання повторюваних операцій.
- Вміти ефективно передавати параметри між функціями.
- Використовувати цикли для роботи з масивами, обчисленням сум та інших задач.

3. Теоретичні відомості

Використання функцій. Функції допомагають структурувати програму, розділяючи її на окремі логічні блоки. Кожна функція виконує певну задачу, що підвищує читабельність коду та спрощує його підтримку.

Приклад оголошення функції:

```
int multiply(int a, int b) {
    return a * b;
}
```

Цикл for. Цикл for використовується для повторення певних дій визначену кількість разів. Його структура наступна:

```
for (ініціалізація; умова; інкремент) {
    // Тіло циклу
}
```

Приклад:

```
for (int i = 0; i < 10; i++) {
    cout << i << endl;
}
```

Приклад програми:

```
#include <iostream>
using namespace std;

// Функція для введення числа
int inputNumber() {
    int n;
    cout << "Введіть число: ";
    cin >> n;
    return n;
}

// Функція для обчислення суми чисел від 1 до n
int calculateSum(int n) {
    int sum = 0;
    for (int i = 1; i <= n; i++) {
        sum += i;
```

```
    }
    return sum;
}

int main() {
    int n = inputNumber();          // Введення числа
    int sum = calculateSum(n);          // Обчислення суми
    cout << "Сума чисел від 1 до " << n << ": " << sum << endl;
    return 0;
}
```

4. Допоміжні конструкції:

Прототип функції – це оголошення функції, яке інформує програму про кількість і тип параметрів, а також тип значення, яке повертає функція. Одним із неймовірно корисних аспектів функцій C++ є створення прототипів функцій. Прототип функції надає інформацію, таку як кількість і тип параметрів і тип значень, що повертаються, щоб пояснити інтерфейс функції для компілятора.

Прототип функції може мати аналогічний вигляд до першого рядка функції (рядок оголошення). Прототип функції вказується на початку програми відразу після підключення бібліотек.

Приклад оголошення прототипів функції:

```
int valAbs (int x);
int greatcd (int a1, int a2);
```

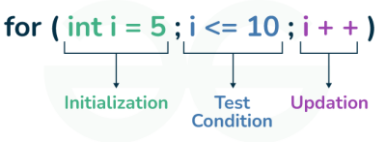
Отже, компоненти прототипу функції такі:

- тип повернення
- назва функції
- список аргументів

У C++ цикл for – це керований входом цикл, який використовується для багаторазового виконання блоку коду для заданого діапазону значень. По суті, цикл for дозволяє повторювати набір інструкцій для певної кількості ітерацій.

Цикл for зазвичай є кращим вибором, ніж цикли while і do-while, якщо кількість ітерацій відома заздалегідь.

Синтаксис циклу for у C++ наступний:



Пояснення частин циклу for :

1. Вираз ініціалізації в циклі for:

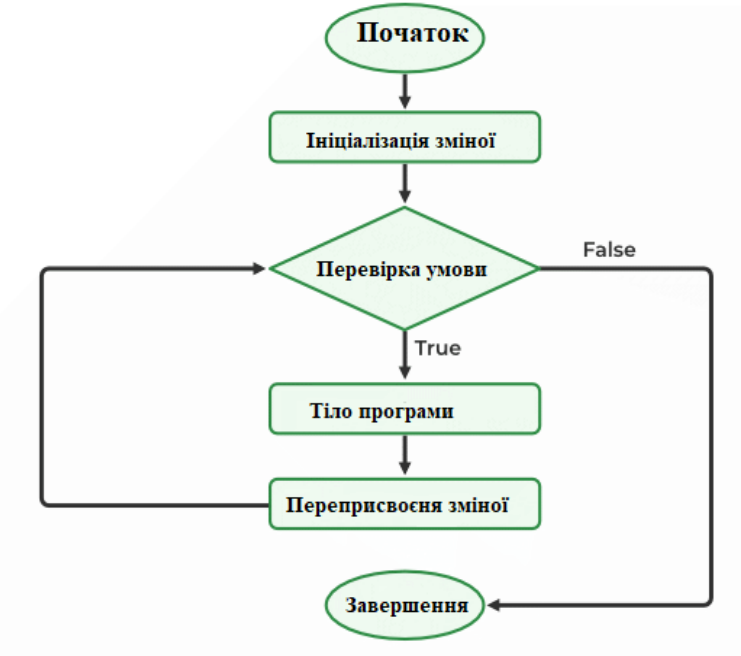
Ми повинні ініціалізувати змінну циклу певним значенням у цьому виразі.

2. Вираз-умова в циклі for:

У цьому виразі ми повинні перевірити умову. Якщо умова оцінюється як істина, тоді ми виконаємо тіло циклу та перейдемо до виразу оновлення. В іншому випадку ми вийдемо з циклу for.

3. Вираз-оновлення у циклі for:

Після кожного виконання тіла циклу цей вираз збільшує/зменшує змінну циклу на деяке значення.



Сценарій виконання циклу for:

1. Керування надається циклу for.
2. Ініціалізація виконана.
3. Потік переходить до стану перевірки Умови.
4. Якщо Умова виконується, то потік надходить у Тіло. В інакшому випадку потік покидає межі циклу.
5. Оператори всередині тіла циклу виконуються.
6. Потік переходить до Оновлення.
7. Процес знову переходить до кроку 3.

5. Завдання

Завдання №1

Відомо, що сейф відкривається при правильному введенні коду з трьох цифр в діапазоні від 0 до 9. Задайте код і потім відкрийте сейф, використовуючи метод перебору за допомогою циклу for.

Додаткові умови:

– має бути підключена кирилиця;

– окрім функції main має бути реалізовано та оголошено як прототипи ще дві функції;

– функція main має знаходитись перед двома іншими функціями;

– перша функція зчитує з клавіатури число і записує його у змінну, якщо число знаходиться в діапазоні від 0 до 9;

– друга функція здійснює перевірку через 3 вкладені цикли for;

– знайдена комбінація записується як число і виводиться не екран;

– має бути лічильник який рахує кількість кроків.

Завдання №2

Виконайте одне додаткове завдання згідно варіанту, співставивши Ваш номер у журналі та номер варіанту у таблиці.

Завдання згідно варіанту

№ варіанту	Завдання
1	**Модифікація завдання:** Додайте функцію для виведення знайденого коду в новому рядку.
2	**Модифікація завдання:** Реалізуйте можливість користувачу задавати тільки 2 цифри).

3	**Модифікація завдання:** Додайте можливість задання коду в шістнадцятковій системі.
4	**Модифікація завдання:** Реалізуйте лічильник, який показує, скільки разів введено невірний код.
5	**Модифікація завдання:** Додайте можливість користувачу вводити не лише цифри, а й букви (a b c).
6	**Модифікація завдання:** Реалізуйте можливість перевірки не одного коду, а декількох комбінацій одночасно (наприклад, користувач вводить кілька можливих кодів).
7	**Модифікація завдання:** Додайте можливість виводити всі комбінації, які перебираються програмою.
8	**Модифікація завдання:** Змініть програму так, щоб вона показувала прогрес (наприклад, кожні 10 комбінацій).
9	**Модифікація завдання:** Додайте можливість користувачу змінювати діапазон для кожної цифри окремо (наприклад, перша цифра від 0 до 5, друга від 6 до 9).
10	**Модифікація завдання:** Реалізуйте додаткову функцію для перевірки випадкових комбінацій (рандомізований пошук коду).
11	**Модифікація завдання:** Додайте можливість користувачу вводити свій коментар після кожної введеної цифри (наприклад, пояснення, чому обрано саме цю цифру).
12	**Модифікація завдання:** Додайте можливість вводити код одразу через пробіл (наприклад, "5 3 7").
13	**Модифікація завдання:** Реалізуйте програму, яка після знаходження коду пропонує користувачу ввести новий

	код і повторює процес.
14	**Модифікація завдання:** Зробіть так, щоб після кількох невдалих спроб користувачу давалася підказка (наприклад, "перша цифра правильна").
15	**Модифікація завдання:** Додайте своє прізвище при знаходженні правильного коду.
16	**Модифікація завдання:** Змініть програму так, щоб код складався з більшого діапазону цифр (наприклад, від 0 до 99 для кожної цифри).
17	**Модифікація завдання:** Додайте функцію, яка запитує у користувача, скільки разів виконати перевірку коду.
18	**Модифікація завдання:** Додайте можливість змінювати кількість цифр у коді динамічно під час виконання програми.
19	**Модифікація завдання:** Додайте можливість задати стартову комбінацію для початку перебору.
20	**Модифікація завдання:** Змініть програму так, щоб вона шукала код у зворотному порядку (від 999 до 000).
21	**Модифікація завдання:** Реалізуйте зворотний алгоритм: користувач вводить знайдений код, а програма виводить, скільки кроків знадобиться, щоб перебрати всі попередні комбінації.
22	**Модифікація завдання:** Додайте можливість користувачу вводити комбінації без перевірки (наприклад, якщо він знає, що правильна комбінація є серед кількох введених).
23	**Модифікація завдання:** Реалізуйте програму так, щоб вона шукала не всі цифри одночасно, а почергово, починаючи з першої цифри.

24	**Модифікація завдання:** Реалізуйте перебір коду в кілька потоків (паралельний пошук).
25	**Модифікація завдання:** Додайте можливість користувачу побачити першу секретну цифру.
26	**Модифікація завдання:** Зробіть так, щоб після кожної спроби програма запитувала у користувача, чи хоче він продовжити пошук.
27	**Модифікація завдання:** Додайте можливість виводити тільки перші 100 комбінацій і зупинити програму після цього.
28	**Модифікація завдання:** Реалізуйте програму так, щоб після кожної спроби виводити останню перевірену комбінацію.
29	**Модифікація завдання:** Додайте можливість змінювати кількість комбінацій, які будуть перевірятися в кожній ітерації (наприклад, перевірити одразу кілька комбінацій).
30	**Модифікація завдання:** Реалізуйте можливість вводу та перевірки не тільки тризначних кодів, але й кодів з довільною кількістю символів.

6. Рекомендації щодо усунення помилок

1) Помилки під час оголошення та виклику функцій.

Неправильний синтаксис оголошення функції:

- Перевірте, чи правильно оголошено функцію (чи вказаний тип поверненого значення, чи правильно передані параметри).
- Переконайтеся, що функція оголошена до її виклику або використовуйте прототип функції.

2) Помилки при використанні циклу for.

Помилка в умові циклу:

- Якщо неправильно налаштована умова циклу, це може призвести до надто великої кількості ітерацій або взагалі відсутності ітерацій.

3) Помилки роботи з типами даних.

Використання неправильного типу даних у функціях:

- Якщо Ви передаєте параметри невідповідного типу, можуть виникати проблеми з обчисленням або компіляцією програми.

Для набуття практичних навичок використовувати матеріали за наступними посиланнями:

Додаткові матеріали

1. https://www.geeksforgeeks.org/cpp-for-loop/
2. https://www.digitalocean.com/community/tutorials/foreach-loop-c-plus-plus
3. https://www.javatpoint.com/cpp-for-loop