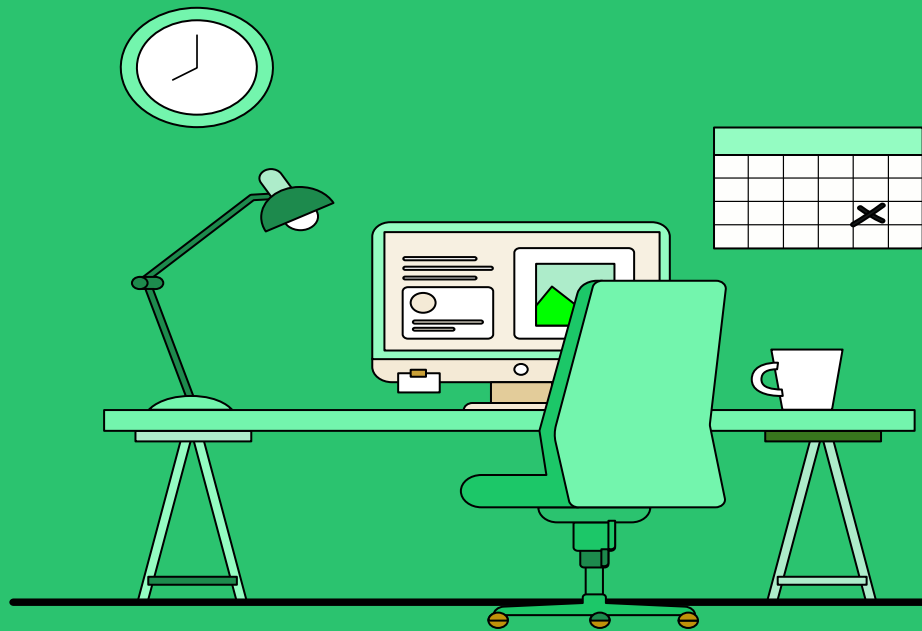


СТРУКТУРНИЙ ПІДХІД

C++

Підсумки
частини 2





The open source, cross-platform IDE

13.12



C++



JavaScript



Java/C#



PHP (Without
MySQL)



Ruby



Pascal



Perl



Lisp



Visual Basic



Haskell



Python



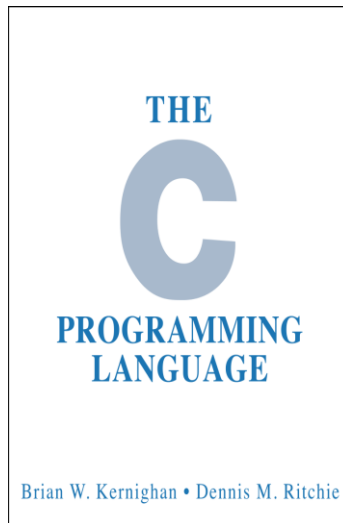
C

Мова програмування C



Денніс Рітчі
Кен Томпсон

#Bell Labs



Мова програмування C++



Б'єрн Страуструп

“Dream no little dreams.”

#Bell Labs

UNIX



LINUX



#Bell Labs



5 min



ЧАС ДЛЯ КОДУ?

Давайте зупинимось і проведемо пошук по
#Bell Labs
#історія мови C
#історія мови C++
#Asteroids ?
#мова B

0:00

Типи даних



Цілі



Дійсні



Символьні

3 базові типи

Виділення пам'яті

Назва	Позначення	Діапазон значень	Розмір (байт)
Цілі типи			
Байт (символ)	char	-128..+127 (будь-який символ ASCII)	1
Байт без знака	unsigned char	0..255	1
Ціле число	Int	-32768..+32767 -2147483648..+ 2147483648	2 4
Коротке ціле	short int	-32768..+32767	2
Коротке ціле без знака	unsigned short int	0..65535 0..4294967295	2 4
Довге ціле	long int	-2147483648..+ 2147483648	4
Довге ціле без знака	unsigned long int	0..4294967295	4

Типи даних

int
double + long

char
string

```
int x;  
int a=34;  
long int b=10000000;  
  
double x;  
long double=12536.65565  
  
char c;  
char ch='$' ;  
string s="Hi!"
```

sizeof()

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

int main() {
    int a;
    cin>>a;
    cout<<sizeof(a);
    return 0;
}
```

int
long int
short int
double
long double
float
char
string
bool

5 min



ЧАС ДЛЯ КОДУ?

Потестіть в коді з попереднього слайду
різні типи даних з модифікаторами long,
short та складіть свою таблицю :
тип- розмір в пам'яті

0:00

Введення - виведення даних

```
#include <iostream>;
```



cin

```
int a;  
cin>>a;
```

```
int a;  
cout<<a;
```



cout

Перша програма



```
#include <iostream>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    cout <<"Hello, world!" ;
```

```
    return 0;
```

```
}
```

Перша програма



```
#include <iostream>
  
int main()
{
    std::cout <<"Hello, world!" ;
    return 0;
}
```

Особливості введення-виведення

cin>>

```
int a,b;  
cin>>a>>b;
```

3 5

3

5

cout<<

```
int a,b;  
cin>>a>>b;  
cout<<a<<endl;  
cout<<b<<endl;  
cout<<a<<"+"<<b<<"="<<a+b;
```

3 + 5 = 8

Особливості виведення дійсних чисел

float - дійсне число ординарної
точності($\pm 3.4 \times 10^{38}$)

double - дійсне число подвійної
точності($\pm 1.7 \times 10^{308}$)

```
#include <iomanip>
```

cin>>

```
float a,b;  
cin>>a>>b;
```

cout<<

```
double a,b;  
cin>>a> b;  
cout<<setprecision(3)<<a;  
cout<<setprecision(2)<<b;
```

Особливості введення-виведення

```
#include <iostream>
using namespace std;
int main()
{
double a,b;
b=rand()*100/100;
cout <<a<<endl;
cout<<b<<endl;
return 0;
}
```



- + **#include <iomanip>**
- + **setprecision()**
- + **fixed**

5 min



ЧАС ДЛЯ КОДУ?

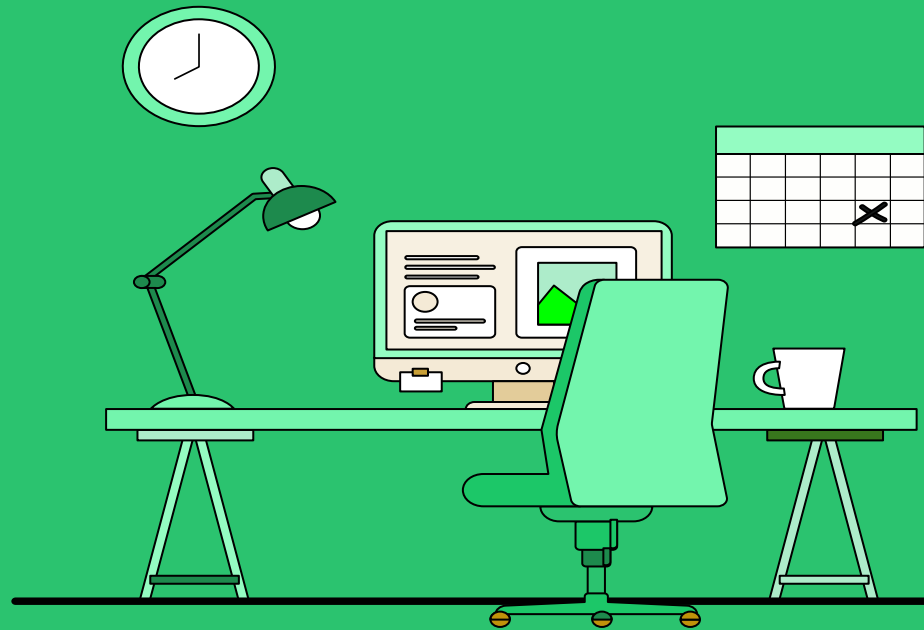
Попрацюємо з дійсними числами:

- Введіть дійсне число та виведіть число з різною точністю
- Згенеруйте дійсне число та виведіть також з різною точністю

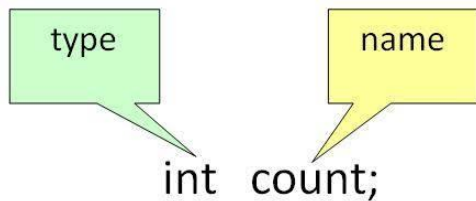
0:00

ВВЕДЕННЯ-ВИВЕДЕННЯ. АРИФМЕТИКА

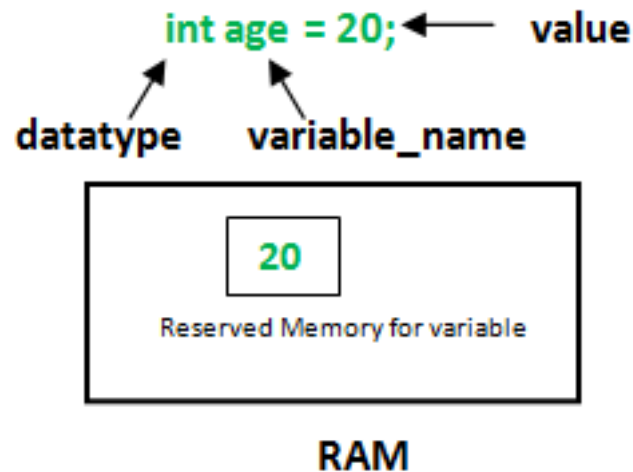
C++



Змінні



Змінна — об'єкт програми, що має ім'я та значення.





Арифметичні дії та розрахункові задачі

```
c= a + b;
```

```
d=a - b;
```

```
l=a*b;
```

```
d=a/2;
```

```
c=a%2;
```

Сума двох чисел

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

```
int main ()
{
    int a,b,c;
    cin>>a>>b;
    c=a+b;
    cout<<c;
    return 0;
}
```



5 min



ЧАС ДЛЯ КОДУ?

Завдання:

1. Напишіть програму розрахунку суми двох чисел з виведенням результату
2. Перевірте для різних значень
3. Поміняйте тип даних та перевірте роботу програми

0:00

Задачі порозрядного ділення



```
int x, d, o;  
cin>>x;  
d=x/10;  
o=x%10;
```

5 min



ЧАС ДЛЯ КОДУ?

Двозначне число поділити на розряди (десятки та одиниці)

0:00



Діалог з
користувачем

Діалог з користувачем

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

int main ()
{
    int a,b,c;
    cout<<"Введіть 2 числа";
    cin>>a>>b;
    c=a+b;
    cout<<a<<"+"<<b<<"="<<c;
    return 0;
}
```



5 min



ЧАС ДЛЯ КОДУ?

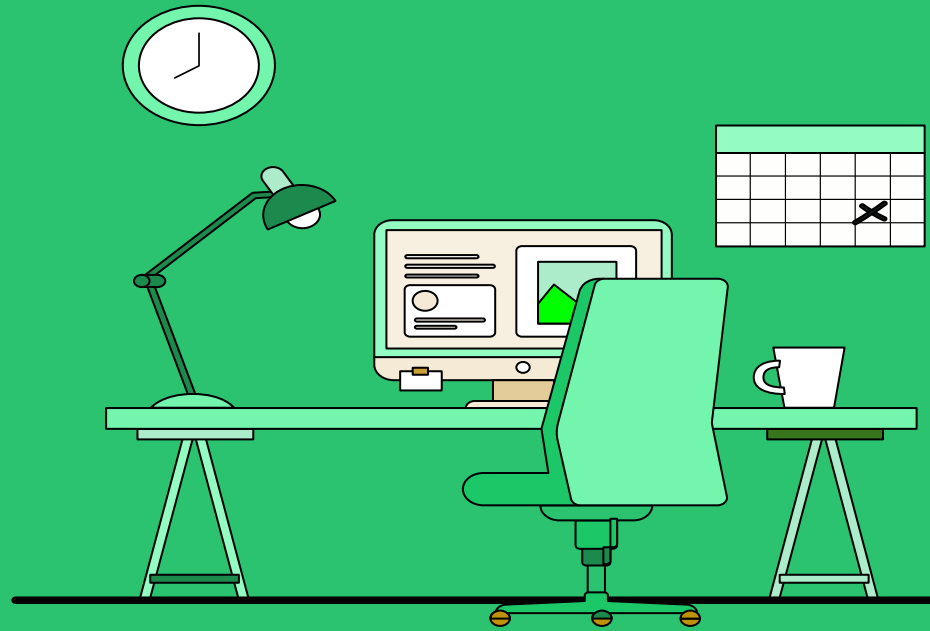
Завдання:

1. Напишіть програму розрахунку суми двох чисел з виведенням результату та виведенням у вигляді прикладу
2. Напишіть код програми "Повторювач", який виводить те саме число, що і було введене
3. Поміняйте тип даних та перевірте роботу програм

0:00

УМОВИ

C++





УМОВИ

- Коротка (if)
- Повна (if - else)
- Складена (декілька умов)

if(умова) дія ;
if(умова) дія 1; дія 2;

Коротка форма



if (умова) дія ;

Якщо умова вірна, то виконується певна дія. яку вказав програміст.

```
if (t>0) cout<<"+";
```


5 min



ЧАС ДЛЯ КОДУ?

Завдання

Спробуємо порахувати

Жили собі люди, які не дуже вміли рахувати.
До 3 в них ще якось виходило, а далі вони вже не могли придумати числа.

Напишіть код, який реалізує їх систему підрахунків.



0:00

A blue rounded square containing the white text "IF".A blue rounded square containing the white text "ELSE".

Повна форма

Якщо умова вірна, то виконується певна дія , інакше - виконується інша.

**if (умова) { дія 1 ; }
else { дія 2; }**

```
if (t>=0)  
    cout<<"+" ;  
else  
    cout<<"-";
```

5 min



ЧАС ДЛЯ КОДУ?

Завдання

Визначити стан за температурою.

Вода може знаходитись у трьох агрегатних станах в залежності від температури: твердому (лід), рідкому (вода) та газоподібному (пар)

Напишіть код, який визначає стан води в залежності від введеної температури



0:00

Логічні операції та складені умови

AND		
IN	IN	OUT
0	0	0
0	1	0
1	0	0
1	1	1

OR		
IN	IN	OUT
0	0	0
0	1	1
1	0	1
1	1	1

NOT	
IN	OUT
0	1
1	0

&& and
|| or
! not

```
if (head==count && time<24) k++;
```

```
if (a<0 || a>100) cout<<"not water";
```

5 min



ЧАС ДЛЯ КОДУ?

Завдання:

- Визначити чи є введене число парним і від'ємним
- Вивести Ок, якщо обидва введених числа парні

0:00

Поговоримо про дужки?

()

Змінюють порядок операцій або використовуються для вхідних параметрів функцій

Приклад: `int main()`

{ }

Блоки коду

Об'єднують частини коду в групи

Приклад: `int main() { ... }`

[]

Визначають індекси масивів

Приклад: `int massiv[100];`

Розглянемо на прикладі

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

int main()
{
    long int a=1000000, b=10;
    if (a>b)
    {
        a=b;
        b=a;
        cout<<a<<" "<<b;
    }
    return 0;
}
```



5 min



ЧАС ДЛЯ КОДУ?

Потестіть код з попереднього слайду, змінюючи параметри та умови

0:00

Тернарна операція

Тернарна операція дозволяє скоротити запис умовної конструкції `if ... else` і найкраще підходить до ситуацій, коли необхідно обрати два різних значення в залежності від однієї умови.

Синтаксис тернарного оператора :

умова ? інструкція 1 : інструкція 1

—

```
result = a > 5 ? a + 4 : a + 5;
```

Оператор вибору

Оператор вибору switch є заміною множинного використання операторів if. Оператор switch порівнює значення однієї змінної з декількома константами.

```
switch (place) {  
    case 1: cout << "GOLD" << endl; break;  
    case 2: cout << "SILVER" << endl; break;  
    case 3: cout << "BRONZE" << endl; break;  
    default: cout << "NO MEDAL" << endl;  
}
```

Всі види умов в C++

IF

ELSE

&&

||
!

1. `if () ;`
2. `if() ; else ;`
3. `if (&& || !). ;`
4. `? : ;`
5. `switch`
{
case:
}

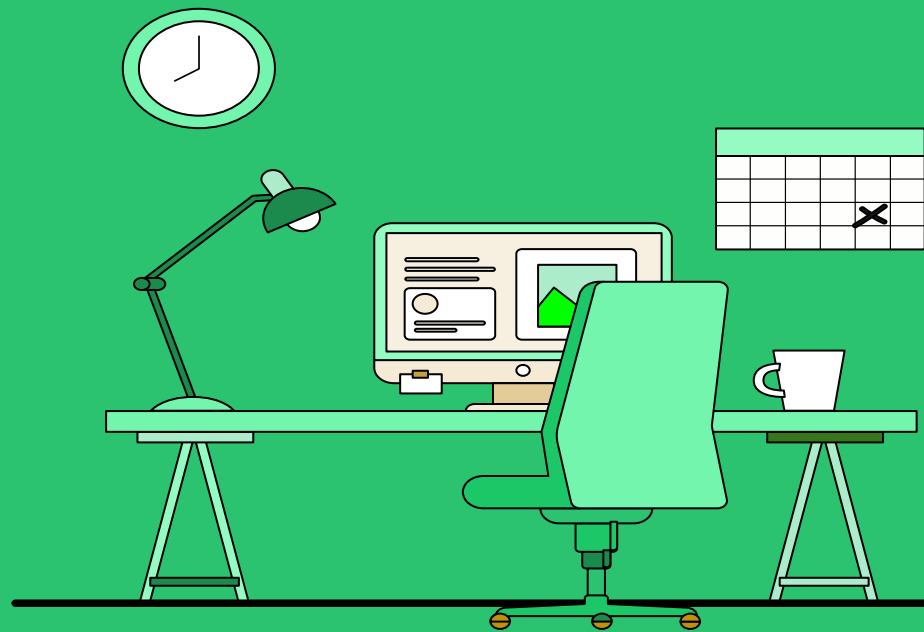
```
switch (place) {  
    case 1: cout << "GOLD" << endl; break  
    case 2: cout << "SILVER" << endl; break  
    case 3: cout << "BRONZE" << endl; break  
    default: cout << "NO MEDAL" << endl;  
}
```

```
result = a > 5 ? a + 4 : a + 5;
```

5

УМОВИ. ПРАКТИКА

C++



Всі види умов в C++

IF

ELSE

&&

||
!

1. `if () ;`
2. `if() ; else ;`
3. `if (&& || !). ;`
4. `? : ;`
5. `switch`
{
case:
}

```
switch (place) {  
    case 1: cout << "GOLD" << endl; break  
    case 2: cout << "SILVER" << endl; break  
    case 3: cout << "BRONZE" << endl; break  
    default: cout << "NO MEDAL" << endl;  
}
```

```
result = a > 5 ? a + 4 : a + 5;
```

5

Правильно прописуємо вкладені умови

```
if();  
    else if();  
        else if();  
else. ;
```

```
if( a == 10 )  
    cout << "Value of a is 10";  
else if( a == 20 )  
    cout << "Value of a is 20";  
else if( a == 30 )  
    cout << "Value of a is 30";  
else  
    cout << "Value of a is not matching";  
cout << "Exact value of a is : " << a;
```

Знак числа

Обмеження по часу: 0.2 секунди
Обмеження по пам'яті: 64 мегабайти

По заданому числу N визначте знак числа.

Формат вхідних даних

У вхідних даних записано одне ціле число N ($-10\,000 \leq N \leq 10\,000$).

Формат вихідних даних

Якщо заданий число додатне, то виведіть квадрат цього числа. Якщо заданий число від'ємне, то виведіть модуль цього числа. Якщо ж задане число – нуль, то і вивести треба нуль.

Приклади

тест	відповідь
5	25
-5	5
0	0

```
if (x>0) cout<<x*x;  
if(x<0) cout<<x*(-  
1);  
if (x==0) cout<<x;
```

Завдання:
перепишіть три короткі умови в одну складену

Непарні Від'ємні

Ограничение по времени: 0.2 секунды
Ограничение по памяти: 64 мегабайта

Вам дано пять целых чисел. Для каждого заданного числа выясните, является ли оно одновременно нечётным и отрицательным.

Формат входных данных

Программа получает на вход сразу пять натуральных чисел. Все числа по абсолютной величине не превосходят 10^9 .

Формат выходных данных

Для каждого числа из входных данных выведите YES, если оно является одновременно нечётным и отрицательным; выведите NO — в противном случае.

Пример

тест	ответ
23	NO
-1	YES
-1234	NO
-12345	YES
-99999	YES

```
if(a<0 && a%2==0) cout<<"YES";
```


Хоча б одна з двох умов

Обмеження по часу: 0.2 секунди
Обмеження по пам'яті: 64 мегабайти

Вам дані п'ять цілих чисел. Для кожного заданого числа з'ясуйте, чи є вірним, що для нього виконується хоча б одна з двох умов:

- це число непарне;
- це число додатне й тризначне.

Формат вхідних даних

Програма отримує на вхід одразу п'ять натуральних чисел. Усі числа за абсолютною величиною не перевищують 10^9 .

Формат вихідних даних

Для кожного числа із вхідних даних виведіть YES, якщо виконується хоча б одна з двох умов, інакше — виведіть NO.

Приклад

тест	відповідь
-21	YES
222	YES
-12	NO
-232	NO
123	YES

Пояснення до прикладу

Для числа -21 виконується перша умова. Для числа 222 виконується друга умова. Для чисел -12 й -232 не виконується жодна з умов, а для числа 123 виконуються обидві умови.

```
if ((x>0 && x>99&& x<1000) || (x%2!=0))
```

5 min



ЧАС ДЛЯ КОДУ?

Завданійня:

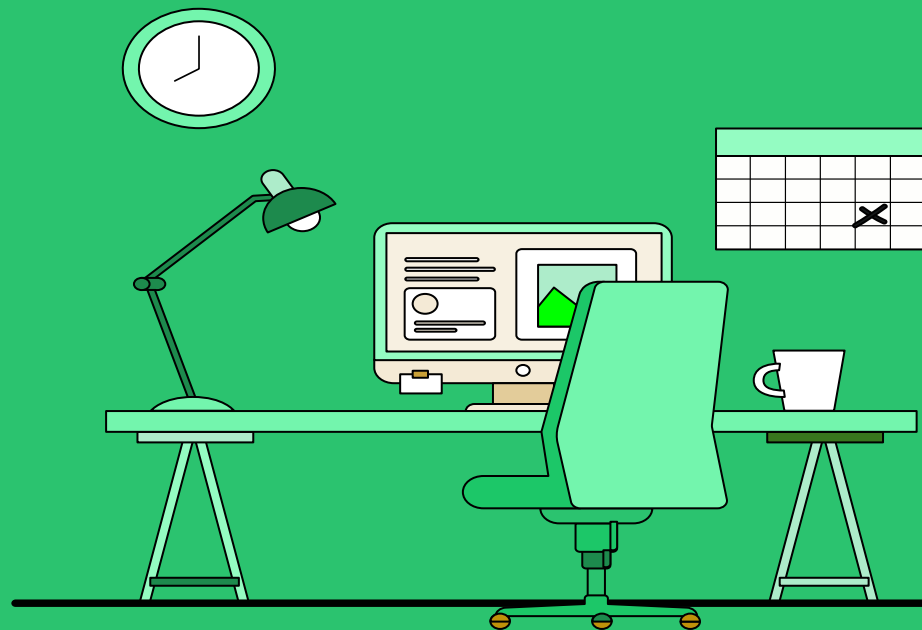
змінити умови в останньому прикладі на

- жодне з 3
- всі 3: умови
- Лише 2 з 3

0:00

ЦИКЛИ

C++



Цикли



for()

Знаємо кількість
повторень

```
for(i=1; i<=10; i++)
```

while()

Знаємо умову, за
якою повторювати

```
i=1;  
while(i<=10)  
{i++;  
}
```

Цикл з умовою

`while()`

{

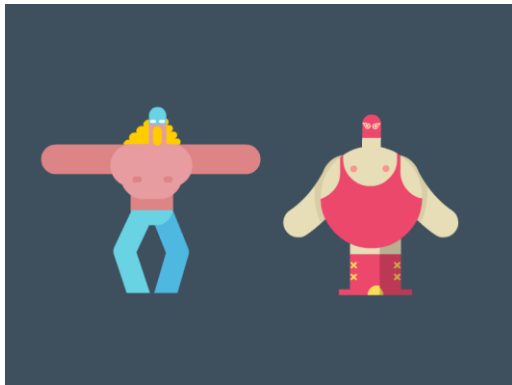
**Повторення дії або блоку дій
поки умова істинна**

}



```
int i=1;
while(i<=3)
{
    i++;
    cout<<i<<" ";
}
```

Цикл з параметром



for (;;) {
Повторення дії або
блоку дій
певну кількість разів
}

```
int i;  
for (i=1; i<=10; i++)  
{  
cout<<i<<" "<<i*i<<endl;  
}
```



do while

```
do  
{  
  cout<<i<<endl;  
  i++;  
}  
while (i<=10);
```

for

while

do while

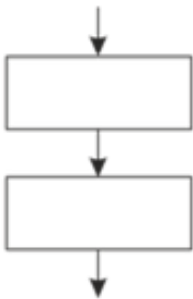
goto



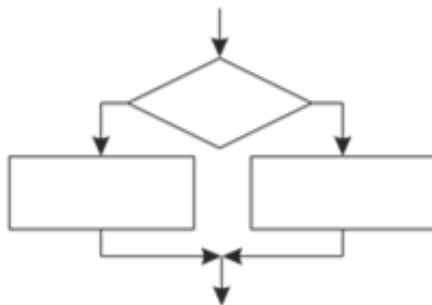
```
#include <iostream>
using namespace std;
int main()
{ int i=1;
do
{
cout<<i<<endl;
i++;
}
while (i<=10);
return 0;
}
```


Алгоритмічні структури

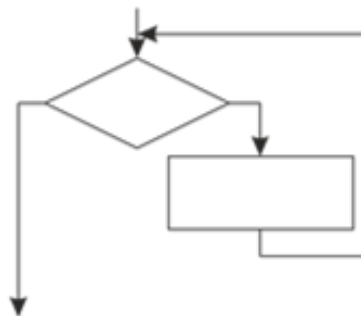
ЛІНІЙНА



УМОВНА



ЦИКЛІЧНА



Масиви

Рядки



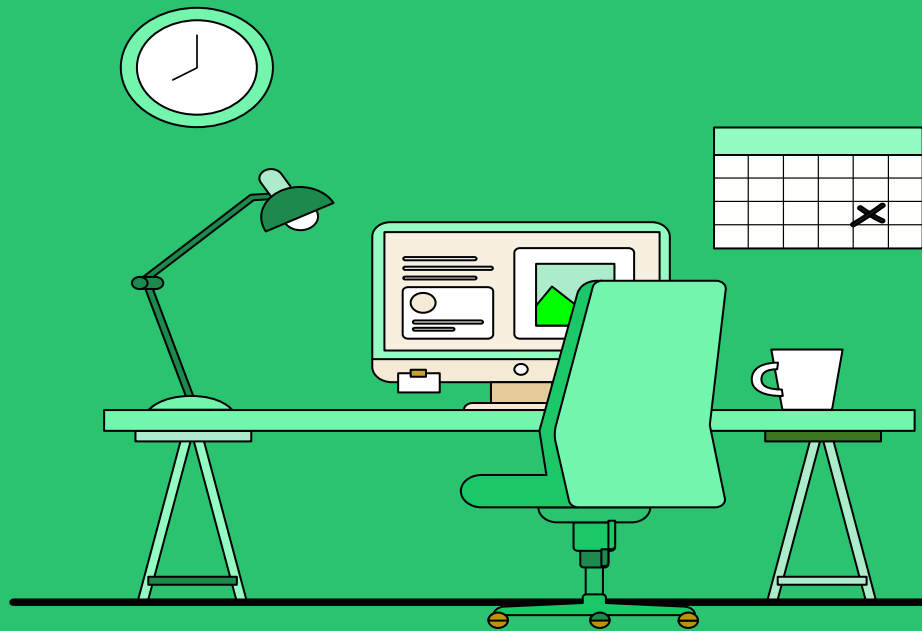
Умови

Цикли

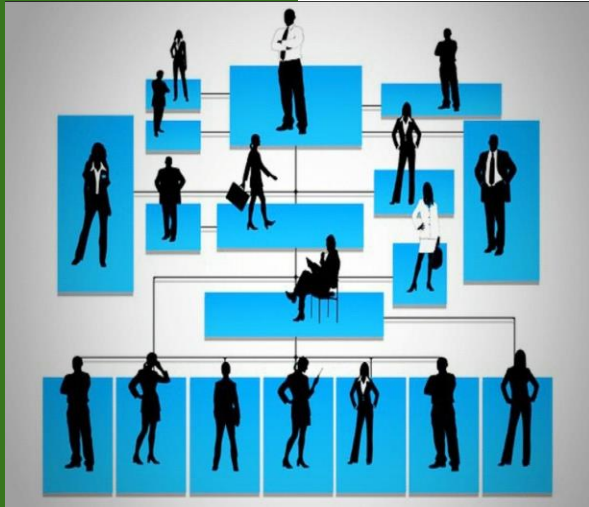
Типи даних. Структура.
Оператори. Операції

ФУНКЦІЇ

C++



Функції

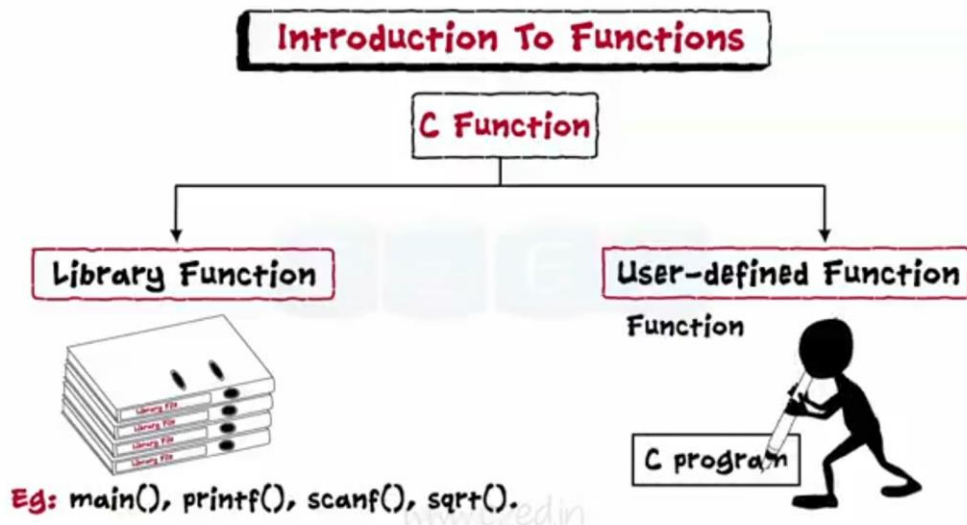


Функції - це блоки коду,
що виконують певні
операції, мають власну
назву

- Бібліотечні
- Користувацькі

Функції в c++

Функції дозволяють зробити програму модульною, тобто розділити програму на кілька маленьких підпрограм (функцій), які виконують частини завдання



Функції в C++

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

int main() {
    int a, b, c;
    cin >> a >> b;
    c=a+b;
    cout << c;
    return 0;
}
```

```
#include <iostream>
using namespace std;
int fout(int c)
{cout <<c;
return 0;
}
int main() {
    int a, b, c;
    cin >> a >> b;
    c=a+b;
    fout(c);
    return 0;
}
```

5 min



ЧАС ДЛЯ КОДУ?

- Потестуйте код з попереднього слайду
- Допишіть виклик функції, яка виводить “The end”

0:00

Виклик функцій за умовою

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

int cat1()
{cout<<"Чорний котик";
  return 0;
}

int main(){
    int x;

    cin >> x;
    if (x<5) cats1();
    else cats2();
    return 0;
}
```

Умови

- Коротка (if)
- Повна (if - else)
- Складена (декілька умов)

if(умова) дія;
if(умова) дія 1; дія 2;

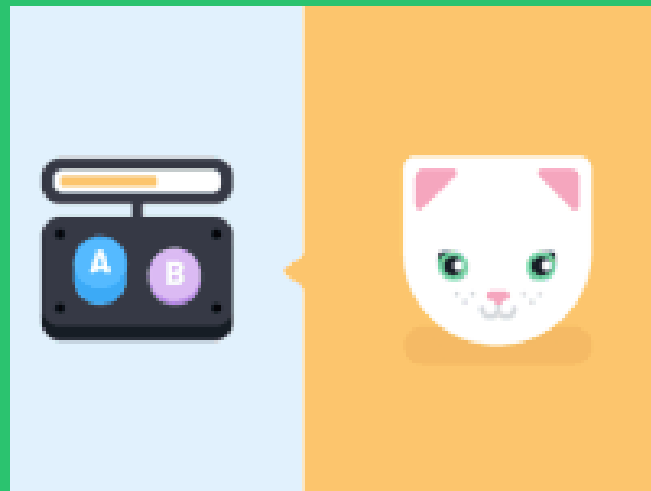


5 min



ЧАС ДЛЯ КОДУ?

Напишіть код з 2 функціями, які викликаються за певною умовою



0:00

Виклик функцій в циклі

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

```
int get_nums()
```

```
{cout<<"Я працюю";
```

```
return 0;
```

```
}
```

```
int main(){
```

```
int x;
```

```
cin >> x;
```

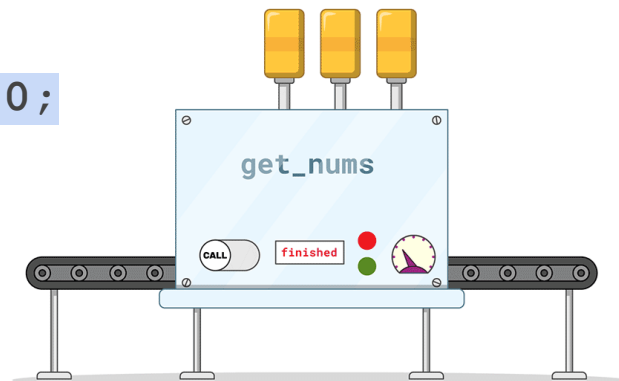
```
while (x<5) {
```

```
    get_nums();
```

```
    x++; }
```

```
return 0;
```

```
}
```

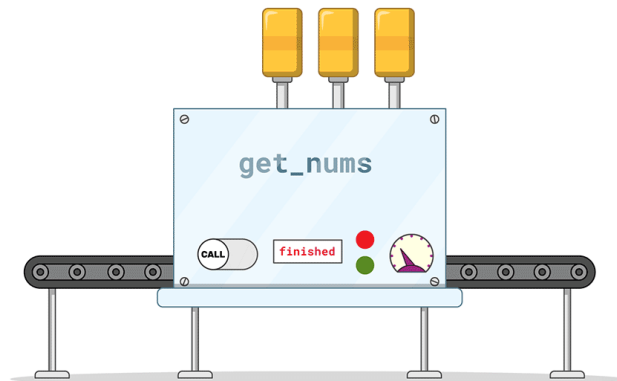


Передаємо дані в функції

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

int get_nums(int x)
{cout<<"Я працюю"<<x<<endl;
  return 0;
}

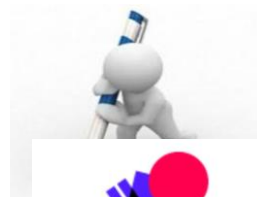
int main(){
    int x;
    cin >> x;
    while (x<5) {
        get_nums(x);
        x++; }
    return 0;
}
```



Оголошення, опис та виклик функції

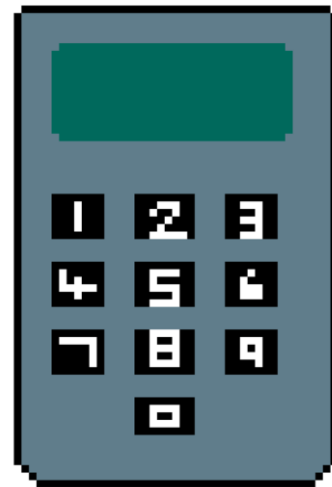
```
#include <iostream>
using namespace std;
int get_nums(int x)
{cout<<"Я працюю"<<x<<endl;
return 0;
}
```

```
int main(){
    int x;
    cin >> x;
    while (x<5) {
        get_nums(x);
        x++; }
    return 0;
}
```



Калькулятор

```
int  sum( int a, int b) {  
    cout<<a+b);  
return 0;  
}  
int main ()  
{int a,b;  
cin>>a>>b;  
sum(a,b);  
return 0;}
```





Параметри функції

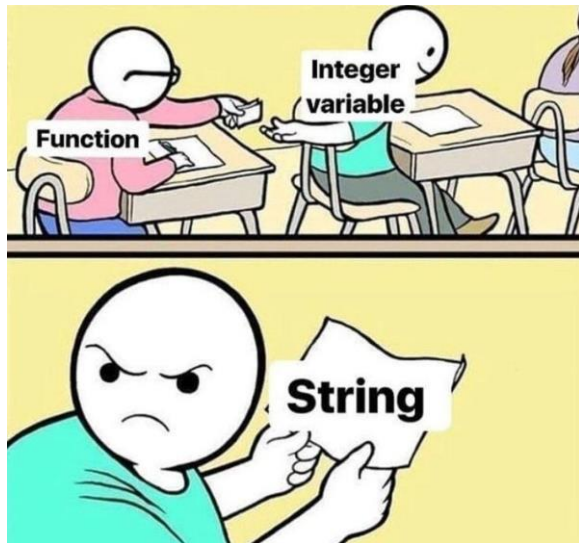
a

b

```
#include <iostream>
using namespace std;
    int sum(int a, int b)
    {
        return a+b;
    }
```

```
int main() {
    int x, y;
    cin >> x>>y;
    cout<<sum(x,y) ;
    return 0;
}
```

Передача параметрів у функції



Потрібно стежити за тим, щоб ви передавали “правильні” дані у функції.

Якщо параметри цілі, то не передавайте дробові числа, чи літери.



Прототип функції

a

b

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

```
int sum(int a, int b) ;
```

```
int main() {
```

```
    int x, y;
```

```
    cin >> x >> y;
```

```
    cout << sum(x, y) ;
```

```
    return 0;
```

```
}
```

```
int sum(int a, int b)
```

```
{
```

```
    return a+b;
```

```
}
```


Що таке return?

Вихід з
функції

Повернення
значення



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

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

```
int main()
{
    int x=10, y=20, z;
    z = mul(x,y);
    cout<< z << endl;
    return 0;
}
```

5 min



ЧАС ДЛЯ КОДУ?

Напишемо код з прототипованою функцією та без прототипу

0:00