

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
імені ТАРАСА ШЕВЧЕНКА**



Програмування на мові C++

Лектор: проф. Васильєв О.М.

Лекція №1.

Знайомство з C++.
Структура програми.
Типи даних, змінні.

Перша програма

```
// Система вводу/виводу:  
#include <iostream>  
// Використовуємо стандартний простір імен:  
using namespace std;  
  
// Головна функція:  
int main() {  
    // Виводимо текст в консоль:  
    cout<<"We are programming in C++!"<<endl;  
    // Пауза перед закриттям консолі:  
    system("PAUSE");  
    // Результат виконання головної функції:  
    return 0;  
}
```

Використання змінних програма

```
#include <iostream>
#include <string>
using namespace std;
int main(){
    // Цілочисельні змінні:
    int age, ThisYear=2013;
    // Текстова змінна:
    string name;
    // Текстова змінна:
    string txt="What is your name?";
    // Виводимо текст:
    cout<<txt<<endl;
    // Зчитуємо текст:
    //cin>>name;
    getline(cin,name);
    // Виводимо текст:
    cout<<"How old are you?"<<endl;
    // Зчитуємо число:
    cin>>age;
    // Рік народження:
    int year=ThisYear-age;
    // Виводимо текст:
    cout<<"OK "+name+", you were born in "<<year<<"!"<<endl;
    // Пауза:
    system("PAUSE");
return 0;}
```

Показати результат

Показати програмний код

Базові типи даних

<i>Ідентифікатор</i>	<i>Тип даних</i>
bool	Логічний тип
char	Символьний тип
wchar_t	Символьний двубайтовий тип
double	Дійсні числа подвійної точності
float	Дійсні числа
int	Цілі числа
void	Значення не повертається

Модифікатори типу

<i>Модифікатор</i>	<i>Пояснення</i>
signed	Значення зі знаком
unsigned	Значення без знака
short	Скорочений тип
long	Розширений тип

Всі чотири модифікатори використовуються з типом **int**

Модифікатори **signed** и **unsigned** використовують з типом **char**

Модифікатор **long** використовують з типом **double**

Мінімальні діапазони даних

(32-розрядне середовище)

<i>Tun</i>	<i>Біти</i>	<i>Діапазон значень</i>
<code>bool</code>	1	Значення <code>true</code> або <code>false</code>
<code>char</code>	8	Від -128 до 127
<code>wchar_t</code>	16	Від 0 до 65535
<code>double</code>	64	Від $2.2\text{E}-308$ до $1.8\text{E}+308$
<code>float</code>	32	Від $1.8\text{E}-38$ до $1.8\text{E}+38$
<code>int</code>	32	Від -2147483648 до 2147483647
<code>unsigned char</code>	8	Від 0 до 255
<code>signed char</code>	8	Від -128 до 127
<code>signed int</code>	32	Від -2147483648 до 2147483647
<code>short int</code>	16	Від -32768 до 32767
<code>unsigned short int</code>	16	Від 0 до 65535

Мінімальні діапазони даних (продовження)

(32-розрядне середовище)

<i>Tun</i>	<i>Біти</i>	<i>Діапазон значень</i>
<code>signed short int</code>	16	Від -32768 до 32767
<code>long int</code>	32	Від -2147483648 до 2147483647
<code>unsigned long int</code>	32	Від 0 до 4294967295
<code>signed long int</code>	32	Від -2147483648 до 2147483647
<code>double</code>	64	Від $2.2\text{E}-308$ до $1.8\text{E}+308$
<code>float</code>	32	Від $1.8\text{E}-38$ до $1.8\text{E}+38$
<code>long double</code>	64	Від $2.2\text{E}-308$ до $1.8\text{E}+308$

Типи операторів в мові C++

- 1. Арифметичні оператори**
- 2. Оператори порівняння**
- 3. Логічні оператори**
- 4. Побітові оператори**

Арифметичні оператори

<i>Оператор</i>	<i>Призначення</i>
+	Додавання
-	Віднімання
*	Множення
/	Ділення. Якщо операндами є цілі числа, виконується ділення націло
%	Залишок від ділення
++	Інкремент
--	Декремент

Скорочені форми операторів:

формат змінна1 оператор=вираз (напр., $x+=z$)

те саме що змінна=змінна оператор вираз ($x=x+z$)

Префіксна і постфіксна форми оп-в інкремента та декремента:

$n++$ або $++n$ ($n=n+1$) $m--$ або $--m$ ($m=m-1$)

Оператори порівняння

Оператор	Призначення
>	Більше
<	Менше
>=	Більше або дорівнює
<=	Менше або дорівнює
==	Дорівнює
!=	Не дорівнює

Оператори присвоєння і рівності:

```
int a=5, b=6;
```

```
a==b; //логічне false
```

```
a=b; //логічне true
```

Логічні оператори

<i>Оператор</i>	<i>Призначення</i>
&&	Логічне І (AND)
 	Логічне АБО (OR)
!	Логічне НІ (NOT)

Логічне виключаюче АБО (XOR) :

A XOR B або

$(A||B)\&\&!(A\&\&B)$

Побітові оператори

Оператор	Призначення
&	Побітове І (AND)
 	Побітове АБО (OR)
^	Побітове виключаюче АБО (XOR)
~	Побітове заперечення НІ (NOT)
>>	Зсув вправо. Старший біт залишається незмінним
<<	Зсув вліво. Молодші біти заповнюються нулями, старші втрачаються

Приклади побітових операцій:

$$5 \& 3 = 1$$

$$5 | 3 = 7$$

$$5 \wedge 6 = 3$$

$$\sim 5 = -6$$

$$5 \gg 2 = 1$$

$$5 \ll 2 = 20$$

Оператор присвоєння і приведення типів

Синтаксис використання: `змінна=вираз`

Особливості:

1. В мові C++ оператор присвоєння повертає значення

```
x=y=z=3;
```

```
n=(m=6)+3;
```

2. Виконується автоматичне приведення типів (тип виразу трансформується до типу змінної, при цьому можуть втрачатись дані)

```
int a;
```

```
double x=3.5;
```

```
a=x; //змінна a отримує значення 3
```

3. Можна виконувати явне приведення типу. Для цього перед виразом у круглих дужках вказують кінцевий тип

```
5/3 ;            //результат 1
```

```
(double)5/3;    //результат 1.66666
```


Тернарний оператор

1. У оператора **3 аргументи**

2. **Синтаксис** виклику: **умова?команда1:команда2;**

Якщо справедлива **умова**, виконується **команда1**. Інакше виконується **команда2**

3. Оператор повертає **результат** – це результат виклику відповідної команди

4. Приклад використання:

```
#include <iostream>
using namespace std;
int main(){
    int a,b;
    cout<<"Enter a = ";
    cin>>a;
    b=a>10?1:0;
    cout<<"b = "<<b<<endl;
    return 0;
}
```