



# Консольний ввід-вивід даних

Лектор:

к.т.н., доцент кафедри прикладної математики

Рижа Ірина Андріївна

# Про що ця лекція

---

- Розглянемо особливості форматowanego вводу-виводу даних у C.
- Опишемо консольний ввід-вивід даних у C++.

# Консольний ввід вивід даних у С

Особливістю сімейства С-подібних мов є відсутність операторів вводу-виводу даних. Замість них використовуються бібліотечні функції.

Ввід-вивід даних на консоль (з клавіатури чи на екран монітора) може бути двох видів:

1. **неформатований** – дає можливість опрацьовувати лише один екземпляр відповідного типу: символу чи стрічки;
2. **форматований** – дозволяє обробити за одне звертання зразу декілька елементів, виконуючи при необхідності перетворення даних:
  - з символьного формату у двійковий – функція вводу **scanf()**;
  - з двійкового представлення у символьне – функція виводу **printf()**.

*Ці функції забезпечуються заголовковим файлом стандартного вводу-виводу **<stdio.h>**.*

# Форматований вивід даних на консоль

Для виведення даних на екран монітора призначена функція `printf()`.

- може приймати довільне число аргументів, з яких перший (обов'язковий) **«форматуюча» стрічка**, яка задає форму перетворення даних із списку виводу.

`printf(форматуюча стрічка, список значень виводу);`

- Виводить на екран відформатовану стрічку, в якій послідовність символів, що починається символом `%` (відсоток), замінюється символами інформації, що призначена для виведення на екран.
- Послідовність символів після `%` розглядають як специфікацію перетворення відповідних даних із списку виводу.

# Форматований вивід даних на консоль

## Форматуюча стрічка

– це взята в лапки послідовність символів, серед яких можуть бути літерали, керуючі символи і власне сам специфікатор формату виводу.

**Специфікатор формату** починається символом %, а далі вказується послідовність обов'язкових і необов'язкових символів, чисел та знаків:

`%[прапорець][ширина][.точність][{h|l|L}] <тип>`

- Необов'язкові складові специфікації форматування – це прапорці, ширина, точність, дескрипторні префікси або модифікатори.
- Обов'язковим символом у специфікації формату є дескриптор типу.

# Форматований вивід даних на консоль

## Дескриптор типу

– це обов'язковий символ у специфікації формату, що визначає, якого типу дане виведеться на екран і в якому вигляді.

i, d	ціле число зі знаком у десятковій системі числення ( <a href="#">int</a> );
u	ціле беззнакове число ( <a href="#">unsigned int</a> );
x, X	вивід цілих беззнакових чисел у шістнадцятковій системі ( <a href="#">hex</a> );
o	вивід цілих беззнакових чисел у вісімковій системі ( <a href="#">oct</a> );
f	вивід даного дійсного типу з фіксованою крапкою у форматі [-]цццц.цццццц;
e, E	вивід даного дійсного типу з плаваючою крапкою у форматі [-]ц.цццццце[+ -]ппппп або [-]ц.ццццццE[+ -]ппппп;
g, G	дійсне число зі знаком чи без у форматі e (E) чи f за вибором системи;
c	вивід символу ( <a href="#">char</a> );
s	вивід послідовності символів.

# Форматований вивід даних на консоль

## Прапорець

– це необов'язковий елемент форматування, який може бути поданий символами:

-	лівобічне вирівнювання даних (за замовчуванням вирівнювання здійснюється вправо);
+	виведення числа зі знаком '+' чи '-' відповідно;
' ' (пробіл)	резервування позиції під знак '-' для від'ємного числа, а для додатного числа замість знаку '+' буде порожня позиція;
#	при виведенні чисел у вісімковій чи шістнадцятковій формі представлення (o, x, X) перед значенням відобразиться 0, 0x, 0X, відповідно.

# Форматований вивід даних на консоль

## Ширина

– необов'язковий параметр форматування, який у вигляді цілого числа вказує на найменшу ширину поля, виділеного для виведення даного, а надлишкові позиції заповнюються пробілами.

- Якщо ширина подається у вигляді нуля (0) – це вказує, що під дане виділено 1 позицію, але ліві надлишкові позиції заповнюватимуться нулями.
- Якщо ширина задається символом \*, то можна мати змінну ширину поля виведення даного.
- Якщо даному замало виділеної зони, то значення ширини ігнорується.



# Форматований вивід даних на консоль

## Точність

– необов'язковий параметр, який для кожного з типів має свій зміст:

- для даних дійсного типу з форматами `f`, `e`, `E` – визначає кількість знаків після десяткової крапки;
- для форматів `g`, `G` – вказує максимальне число значущих цифр, що будуть виведені;
- для цілих у десятковій системі числення – встановлює мінімальне число цифр, якими має бути зображене число;
- для формату `s` – вказує на кількість символів, які будуть виведені з даної послідовності символів;

*Якщо точність є нульовою (.0), то дійсне число виведеться без крапки.*

*Якщо кількість розрядів цілого числа є менша за цей параметр, тоді його ліві розряди заповнюються нулями, що не змінює величини самого числа.*

# Форматований вивід даних на консоль

---

## Модифікатор h|l|L

– необов'язковий параметр, який визначає чи дане, що виводиться буде типу `short`, `long` чи `long double`, відповідно.

*Форматуюча стрічка може також включати символи керуючої послідовності, розділювачі тощо.*

# Форматований ввід даних з консолі

Для введення даних з клавіатури використовують функцію `scanf()`.

- Приймає з консолі (клавіатури) усі символи, що були введені до натискання клавіші ENTER, і вносить їх у буфер.

`scanf(форматуюча стрічка, список значень вводу);`

- За форматуючою стрічкою визначає спосіб перетворення введених символів у двійковий формат згідно заданих специфікацій.
- Одержане в наслідок перетворень двійкове число заноситься за адресою змінної, яка є у списку адрес.

# Форматований ввід даних з консолі

## Форматуюча стрічка

має структуру подібну до форматної стрічки функції форматованого виведення `printf()`:

`%[*][ширина][{h|l|L}] <тип>`

## Дескриптор типу

– обов'язковий параметр, який визначає, якого типу дане вводиться, тобто, до якого типу слід перетворити введену послідовність символів.

- Символи для дескриптора типу є аналогічними, як і для функції `printf()`.

# Форматований ввід даних з консолі

## Необовязкові модифікатори

*	відміна вводу значення за даним форматом;
	Наприклад "%*d"
ширина	ціле невід'ємне число, що вказує на кількість символів, які повинні бути введеними;
	Наприклад "%10s"
h l L	дане зчитується як тип <b>short</b> (hd/hu), <b>long</b> (ld/lu), <b>long double</b> (Lf/Le).

# Консольний ввід-вивід даних у C++

У C++ частіше використовується ввід-вивід, який забезпечується за допомогою заголовкового файлу `<iostream.h>`.

- З точки зору програми ввід-вивід трактується як потік байт, які є представленням відповідних символьних або числових даних.
- Сам процес вводу-виводу ефективно реалізується за допомогою буферів, що дозволяє вирівняти швидкості передачі даних між оперативною пам'яттю та зовнішнім носієм.

# Консольний вивід даних у C++

## Об'єкт `cout`

призначений у C++ для роботи зі стандартним потоком виводу (зазвичай це вивід на екран монітору).

- Використовується з операцією вставки `<<`, яка виводить потік байтів у буфер виводу.
- Дозволяє розпізнати всі базові типи мови C++ (`char`, `int`, `float`, `double`, `long double`, тощо).
- Операція вставки `<<` перетворює значення у текстовий вид і може бути **каскадовою**, тобто її можна використати кілька разів з одним об'єктом `cout`. Це дозволяє проводити конкатенацію виводу

```
cout << x << y - z << "символьна стрічка";
```

# Форматування виводу у cout

За замовчуванням перетворені у символьні стрічки дані форматуються наступним чином:

- дані типу `char` виводяться як один символ;
- числові дані цілого типу виводяться як десяткові цілі з виводом знаку ‘-’ для від’ємних чисел;
- стрічкові дані (послідовності символів) виводяться в полі, довжина якого еквівалентна кількості символів у стрічці;
- дані дійсного типу виводяться у вигляді шести десяткових розрядів, за якими може слідувати порядок.



# Форматування виводу у cout

## Функції маніпулятори

- задають систему числення

`hex(cout);`                      `dec(cout);`                      `oct(cout);`

або

`cout << hex;`                      `cout << dec;`                      `cout << oct;`

- задають ширину поля виводу

`cout.width(n)` – яка задає ширину поля виводу (діє одноразово);

- заповнюють невикористані системою позиції виводу вказаним символом (у стандартному режимі роботи незаповнені позиції виводу заповнюються символами інтервалу)

`cout.fill('символ')`

- встановлювати число розрядів при виводі дійсних чисел

`cout.precision(n)` – по замовчуванню задає загальне число цифр, проте у фіксованому та експонентному форматах – задає число цифр після десяткової крапки.

# Консольний ввід даних у C++

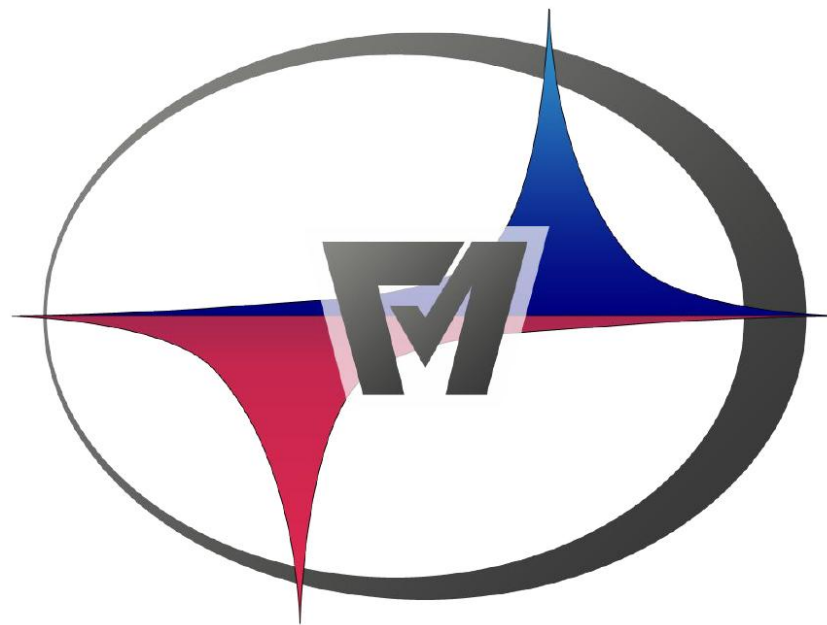
## Об'єкт cin

призначений у C++ для роботи зі стандартним потоком вводу (зазвичай це ввід з клавіатури).

- Використовується з операцією взяття з потоку >>, яка бере, дані з потокового об'єкта, що стоїть зліва від неї, і записує їх у змінну, що є праворуч.

```
cin >> value;
```

- Використовуючи потоковий об'єкт cin та операцію взяття з потоку >>, можна надавати значення змінним різних типів, але ці змінні мають бути попередньо оголошеними.
- Операція >> може бути **каскадовою**, тобто її можна використати кілька разів з одним об'єктом cin.



Кафедра прикладної математики

<http://amath.lp.edu.ua>