

## Streamy (proudy, roury)

Streamy jsou nový způsob vstupů a výstupů zavedený v jazyce C++.

### Streamy pro vstup ze standardního zařízení a výstup na standardní zařízení

Jsou začleněny do standardního jmenného prostoru `std`. Při použití streamů musíme buďto uvést, že používáme standardní jmenný prostor

```
using namespace std;
```

nebo v každém zápisu streamu uvést pomocí operátoru rozlišení `::` jeho příslušnost ke standardnímu jmennému prostoru:

```
std::cout << "C++";
```

### Streamy

<b>cin</b>	standardní vstupní stream (C: stdin)
<b>cout</b>	standardní výstupní stream (C: stdout)
<b>cerr</b>	standardní stream výstupu chybových zpráv (C: stderr)

Přetížení operátoru (operator overloading): Je specifickým druhem polymorfismus. Operátor má různou funkci pro různé typy operandů.

Pro výstupní streamy se používá přetížení operátoru `<<`:

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

cout << výraz;

cout << 'C' << "++"; // C++

int i=2;

cout << i << '+' << 1 << " je " << i+1; // 2+1 je 3

cout << "3^5 = " << (3^5); // 3^5 = 6

char c=65;

cout << c << ' ' << (int)c; // A 65
```

Pro vstupní streamy se používá přetížení operátoru `>>`:

```
unsigned u;

float x;

cin >> u >> x; // z klávesnice: 5 3.14

cout << u << " " << x; // 5 3.14
```

## Formátování vstupů a výstupů – manipulátory

<b>endl</b>	vloží do výstupu nový řádek
-------------	-----------------------------

Následující přehled ukazuje manipulátory. Platnost manipulátoru začíná jeho uvedením v streamu a končí uvedením takového manipulátoru, který jeho platnost mění.

<b>dec</b>	dekadická soustava
<b>hex</b>	hexadecimální soustava
<b>oct</b>	oktalová soustava

```
cout << dec << 10 << endl    // 10
      << hex << 10 << endl    // a
      << oct << 10 << endl;    // 12

cout << 10 << endl;           // 12
```

<b>showbase</b>	zobrazí označení číselné soustavy	<b>nshowbase</b>
-----------------	-----------------------------------	------------------

```
cout << showbase << dec << 10 << endl    // 10
      << hex << 10 << endl    // 0xa
      << oct << 10 << endl;    // 012
```

<b>showpos</b>	zobrazí znak + u nezáporných čísel	<b>nshowpos</b>
----------------	------------------------------------	-----------------

```
cout << showpos << 0 << endl    // +0
      << 1 << endl    // +1
      << nshowpos << 2 << endl;  // 2
```

<b>uppercase</b>	velká písmena	<b>nouppercase</b>
------------------	---------------	--------------------

```
cout << uppercase << showbase << hex << 200 << endl    // 0XC8
      << nouppercase << 200 << endl;    // 0xc8
```

<b>setw</b>	nastaví šířku výstupu
-------------	-----------------------

```
cout << setw(5) << 10 << endl    //    10
      << 10 << endl;    // 10

cout << setw(5) << 1.2 << endl    //    1.2
      << 1.2 << endl;    // 1.2

cout << setw(5) << 123456789 << endl;    // 123456789
```

<b>left</b>	zarovnání vlevo
<b>right</b>	zarovnání vpravo
<b>internal</b>	vložení výplňkových znaků dovnitř

```
cout << setw(5) << left << -10 << endl // -10
    << setw(5) << right << -10 << endl // -10
    << setw(5) << internal << -10 << endl; // - 10
```

<b>setfill</b>	nastaví výplňkový znak
----------------	------------------------

```
cout << setfill('*') << setw(10) << 1000 << endl; // *****1000
cout << setfill('0') << setw(10) << 1234 << endl; // 0000001234
```

<b>setprecision</b>	nastaví přesnost
---------------------	------------------

```
cout << setprecision(5) << 3.14159 << endl; // 3.1416
cout << setprecision(5) << 3.14 << endl; // 3.14
cout << setprecision(5) << 123456.789 << endl; // 1.2346E+005
```

<b>fixed</b>	nastavení přesnosti se vztahuje na desetinná místa
--------------	--

```
cout << setprecision(4) << fixed << 3.14159 << endl; // 3.1416
cout << setprecision(4) << fixed << 3.14 << endl; // 3.1400
```

<b>scientific</b>	zobrazí číslo v semilogaritmickém tvaru
-------------------	---

```
cout << scientific << 123.4 << endl; // 1.234000e+002
cout << setprecision(2)
    << scientific << 123.4 << endl; // 1.23e+002
```

<b>boolalpha</b>	vypíše hodnoty typu <b>bool</b> klíčovými slovy	<b>noboolalpha</b>
------------------	---	--------------------

```
bool b = true;

cout << b << endl // 1
    << boolalpha << b << endl; // true
```

<b>skipws</b>	přeskočí na vstupu „bílé“ znaky (whitespaces)	<b>noskipws</b>
---------------	---	-----------------

Implicitně je nastavena volba **skipws**.

```
char a,b,c,d,e,f;

cin >> a >> b >> c >> noskipws >> d >> e >> f;

cout << a << b << c << d << e << f << endl;
```

Vstup:  
123

456

Výstup:

123

4

<b>ws</b>	na aktuální pozici vstupu odstraní všechny „bílé znaky“
-----------	---

```
char a,b,c,d,e,f;
```

```
cin >> noskipws >> a >> b >> c >> ws >> d >> e >> f;
```

```
cout << a << b << c << d << e << f << endl;
```

Vstup:

1 2 3 4 5 6

Výstup:

1 23 4

## Manipulátory používané při zápisu do souboru

<b>ends</b>	vloží do výstupu znak <code>'\0'</code>
-------------	---

<b>unitbuf</b>	vyrovnávací paměť je vyprázdněna po každém vložení	<b>nounitbuf</b>
----------------	--	------------------

<b>flush</b>	vyrovnávací paměť je vyprázdněna po vložení
--------------	---