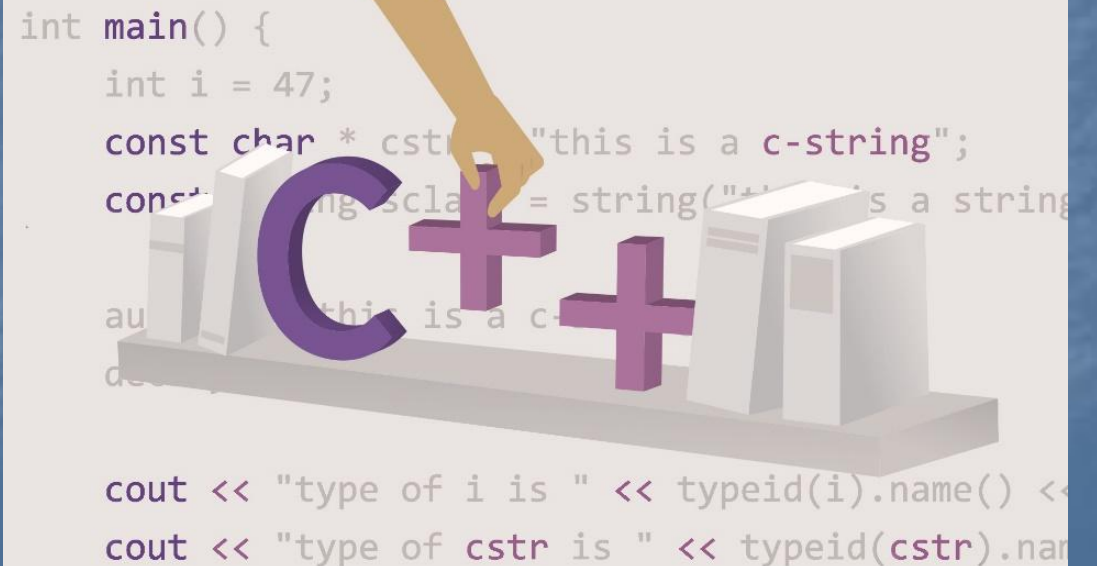


Строки в языке C++

string



The image shows a hand pointing to the C++ logo, which is a purple 'C' followed by two purple '+' signs. The logo is placed on a shelf with several books. The background is a light gray with faint C++ code snippets.

```
int main() {  
    int i = 47;  
    const char * cstr = "this is a c-string";  
    const string scla = string("this is a string");  
    auto this is a c-  
    de  
  
    cout << "type of i is " << typeid(i).name() <<  
    cout << "type of cstr is " << typeid(cstr).nar
```

Символьный тип данных

Описание

```
char b, c;
```

Присваивание значения

```
b = '#';  
c = 65;
```

Сравнение

```
if (b > c) ...
```

Строки в языке C++

В C++ строки были добавлены как объекты класса **string**

Для этих объектов:

определен оператор конкатенации (+),

- оператор присваивания (=)
- при работе с такими строками автоматически проходит проверка переполнения строки и увеличение ее размера при необходимости.

Библиотека

```
#include <string>
```

Объявление

```
string s1, s2;
```

Ввод

```
cin >> s1;           //до пробела  
getline(cin, st);    //с пробелами
```

Присваивание значения

```
S = "Привет, ";
```

Использование конкатенации

```
S = s1 + s2;
```

Вывод

```
cout << S << endl;
```

Использование сравнения

```
if (s1 > s2) ...
```

Первый и последний символ

```
char x = s.front();  
char x = s.back();
```

Добавление, удаление в конце строки

```
s.push_back(char x);           s += x;  
s.append(string q);           s += q;  
s.pop_back();                  s.erase(s.back(), 1);
```

Поиск в строке

вычисление длины строки

```
n = S.length();  
n = S.size();
```

проверка является ли строка пустой

```
S.empty();
```

обмен строк

```
S.swap(s1);
```

выделение подстроки из k символов,
начиная с позиции n

```
string q = S.substr(n, k);
```


ПОИСК ВХОЖДЕНИЯ СЛЕВА

```
S.find(s1);      //строки s1  
S.find(c);      //символа c
```

ПОИСК ВХОЖДЕНИЯ СПРАВА

```
S.rfind(s1);    //строки s1  
S.rfind(c);    //символа c
```

Если значение не найдено
возвращает **-1** или (что тоже самое)

string::npos

Находит первый символ строки, равный одному из символов в переданной последовательности символов

```
s.find_first_of(string q);
```

Находит первый символ, не равный ни одному символу из переданной последовательности символов

```
s.find_first_not_of(string q);
```

Аналогично для последнего символа

```
s.find_last_of(string q);  
s.find_last_not_of(string q);
```

Находит первый символ строки, равный одному из символов в переданной последовательности символов. Поиск начинается с позиции `pos`, т.е. найденный символ не может находиться в позиции, предшествующей `pos`.

```
s.find_first_of(string q, int pos);
```

Аналогично для всех поисковых функций, можно начинать с позиции `pos`

Преобразование типов (работает начиная с C++ 14)

Преобразует строку в целое число

```
stoi, stol, stoll
```



Преобразует строку в целое беззнаковое число

```
stoul, stoull
```

Преобразует строку в действительное число

```
stof, stod, stold
```

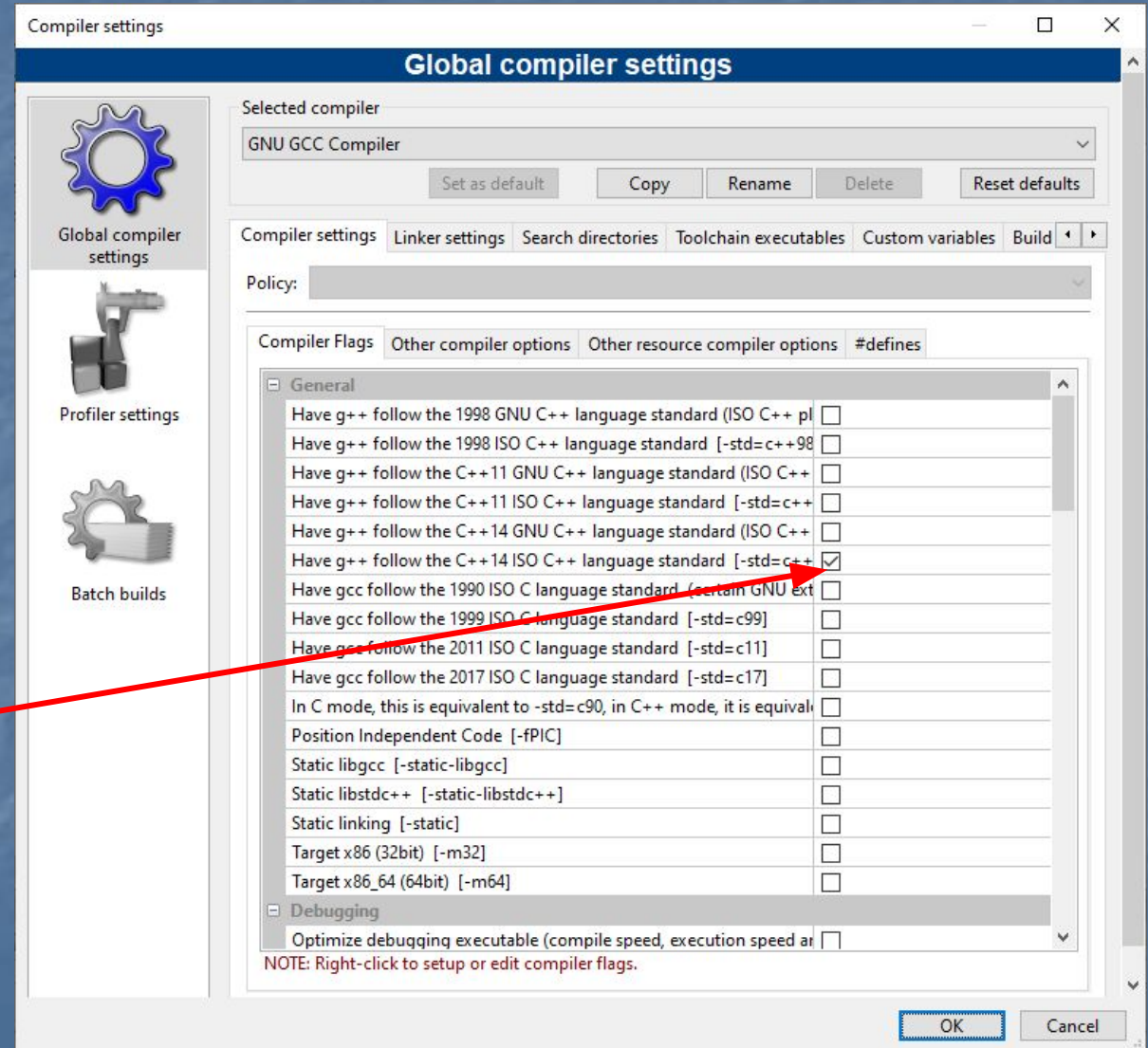
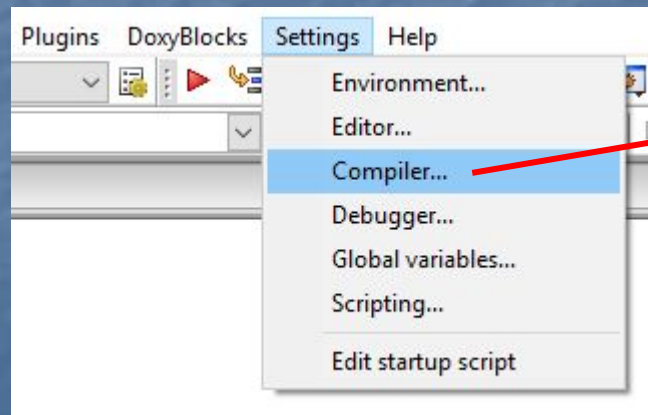
Преобразует число в строку

```
to_string
```

Преобразование типов

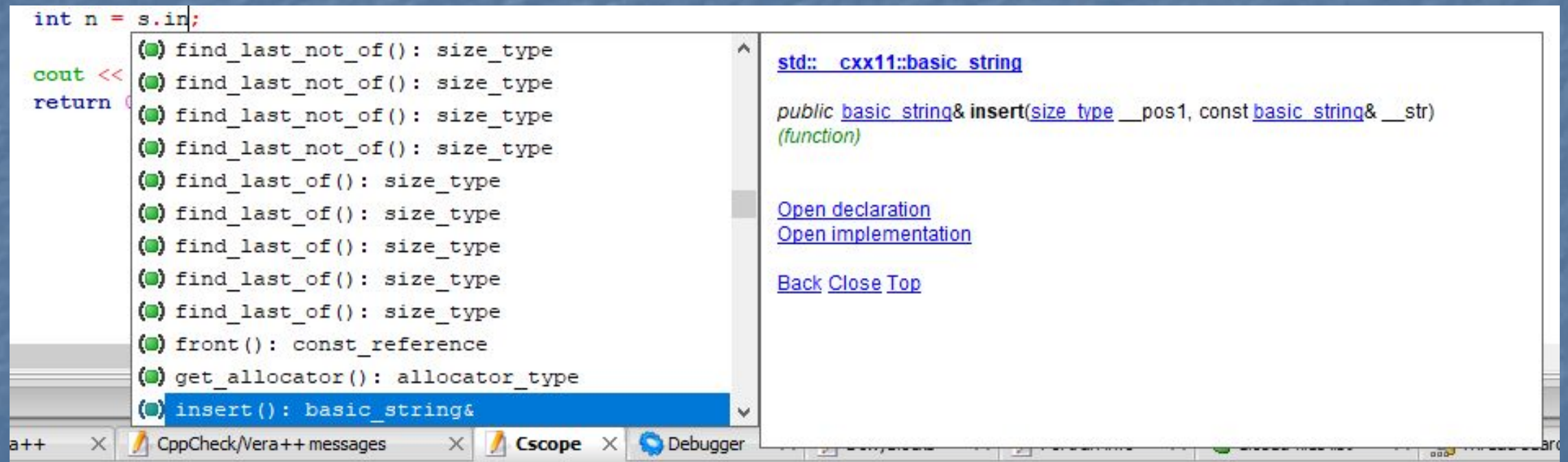
```
int main()
{
    string s1,s2;
    cin >> s1 >> s2;
    int x1 = stoi(s1);
    int x2 = stoi(s2);
    cout << x1 + x2 << endl;
    cout << x1 / x2 << endl;
    double y1 = stod(s1);
    double y2 = stod(s2);
    cout << y1 + y2 << endl;
    cout << y1 / y2 << endl;
    return 0;
}
```

Преобразование типов (работает начиная с C++ 14)



Подсказка имен функций

```
using namespace std::__cxx11;
```



вставка в позицию n строки s1

```
S.insert(n, s1);
```

вставка в позицию n1 k символов строки s1, начиная с позиции n2

```
S.insert(n1, s1, n2, k);
```

удаление k символов строки S, начиная с позиции n

```
S.erase(n, k);
```


замена k символов строки S,
начиная с позиции n на строку s1

```
S.replace(n, k, s1);
```

замена k1 символов строки S, начиная с позиции n1,
на k2 символов строки s1, начиная с позиции n2

```
S.replace(n1, k1, s1, n2, k2);
```

Примеры использования функций

Исходные данные	Команда	Результат
<code>st = "informatika"</code>	<code>a=st.substr(2, 5);</code>	<code>a = "forma"</code>
<code>st = "informatika"</code>	<code>st.erase(2, 7);</code>	<code>st = "inka"</code>
<code>st = "vina'"</code> <code>a = "elich"</code>	<code>st.insert(1, a);</code>	<code>st = "velichina"</code>
<code>st = "informatika"</code> <code>a = "sudeistvo"</code>	<code>st.replace(2, 7, a, 2, 3);</code>	<code>st = "indeika"</code>

Подробнее о string

http://ru.cppreference.com/w/cpp/string/basic_string

http://scrutator.me/post/2014/09/02/cpp_strings_basics.aspx

