Overloading Ostream Operator ☆

33/44 challenges solved

Rank: 5729 | Points: 640

Your Overloading Ostream Operator submission got 20.00 points. Share Tweet

Try the next challenge | Try a Random Challenge

Problem Submissions Leaderboard Editorial

The task is to overload the << operator for Person class in such a way that for p being an instance of class Person the result of:

```
std::cout << p << " " << <some_string_value> << std::endl;
```

produces the following output:

```
first_name=<first_name>,last_name=<last_name> <some_string_value>
```

where:

- `<first_name>` is the value of p's `first_name_`
- `<last_name>` is the value of p's `last_name_`
- `<some_string_value>` is an arbitrary `std::string value`

Input Format

The input is read by the provided locked code template. In the only line of the input there are 3 space-separated strings `first_name`, `last_name`, `event`. The values of `first_name` and `last_name` will be used to create an object p of type Person. The value of `event` will be used by the provided code to produce the output.

Constraints

- Each word in the input contains only English letters and is no longer than 15 characters

Output Format

The output should be produced by the provided locked code template. This code will use the implementation of Person public methods and the overloaded << operator to produce the output. Specifically, the output wiil be produced by the following code:

```
cout << p << " " << event << endl;
```

Sample Input 0

```
john doe registered
```

Sample Output 0

```
first_name=john,last_name=doe registered
```

C++14

```
#include <iostream>

using namespace std;

class Person {
public:
    Person(const string& first_name, const string& last_name) : first_name_(first_name), last_name_(last_name) {}
    const string& get_first_name() const {
      return first_name_;
    }
    const string& get_last_name() const {
      return last_name_;
    }
private:
    string first_name_;
    string last_name_;
};
#include <type_traits>
template <typename Type>
auto operator<<(std::ostream &out, const Type &person)
    -> std::enable_if_t<std::is_same<Person, Type>::value, std::ostream &> {
  return out << "first_name=" << person.get_first_name()
             << ",last_name=" << person.get_last_name();
}

int main() {
    string first_name, last_name, event;
    cin >> first_name >> last_name >> event;
    auto p = Person(first_name, last_name);
    cout << p << " " << event << endl;
    return 0;
}
```

Line: 18 Col: 23

Upload Code as File □ Test against custom input

Run Code Submit Code

You have earned 20.00 points!

33/44 challenges solved.

75%

C++
CPP
★★★★★

Congratulations

You solved this challenge. Would you like to challenge your friends?

Next Challenge

- Test case 0 ✓
- Test case 1 ✓
- Test case 2 ✓
- Test case 3 ✓
- Test case 4 ✓
- Test case 5 ✓
- Test case 6 ✓

Compiler Message

```
Success
```

Input (stdin)

Download

```
john doe registered{-truncated-}
```

Download to view the full testcase

Expected Output

Download

```
first_name=john,last_name=doe registered{-truncated-}
```

Download to view the full testcase

Contest Calendar | Blog | Scoring | Environment | FAQ | About Us | Support | Careers | Terms Of Service | Privacy Policy | Request a Feature