‹ Introduction Challenges

Virtual Functions

by saikiran9194

Problem Submissions Leaderboard Discussions

This problem is to get you familiar with virtual functions. Create three classes *Person, Professor* and *Student*. The class *Person* should have data members name and age. The classes *Professor* and *Student* should inherit from the class *Person*.

The class *Professor* should have two integer members: *publications* and cur_id. There will be two member functions: *getdata* and *putdata*. The function *getdata* should get the input from the user: the *name, age* and *publications* of the professor. The function *putdata* should print the *name, age, publications* and the cur_id of the professor.

The class *Student* should have two data members: *marks*, which is an array of size 6 and cur_id. It has two member functions: *getdata* and *putdata*. The function *getdata* should get the input from the user: the *name, age*, and the *marks* of the student in 6 subjects. The function *putdata* should print the *name, age*, *sum* of the marks and the cur_id of the student.

For each object being created of the *Professor* or the *Student* class, sequential id's should be assigned to them starting from 1.

Solve this problem using virtual functions, constructors and static variables. You can create more data members if you want.

Note: Expand the main function to look at how the input is being handled.

Input Format

The first line of input contains the number of objects that are being created. If the first line of input for each object is 1, it means that the object being created is of the *Professor* class, you will have to input the *name, age* and *publications* of the professor.

If the first line of input for each object is 2, it means that the object is of the *Student* class, you will have to input the *name, age* and the *marks* of the student in 6 subjects.

Constraints

$1 \leq len_{name} \leq 100$, where len_{name} is the length of the name.
$1 \leq age \leq 80$
$1 \leq publications \leq 1000$
$0 \leq marks \leq 100$, where marks is the marks of the student in each subject.

Output Format

There are two types of output depending on the object.

If the object is of type *Professor*, print the space separated *name, age, publications* and *id* on a new line.

If the object is of the *Student* class, print the space separated *name, age*, the *sum of the marks* in 6 subjects and *id* on a new line.

Sample Input

```
4
1
Walter 56 99
2
Jesse 18 50 48 97 76 34 98
2
Pinkman 22 10 12 0 18 45 50
1
White 58 87
```

Sample Output

```
Walter 56 99 1
Jesse 18 403 1
Pinkman 22 135 2
White 58 87 2
```

f 🐦 in

Submissions: 17967
Max Score: 40
Difficulty: Medium

Rate This Challenge:
☆☆☆☆☆

More

Current Buffer (saved locally, editable)

C++

```cpp
#include <cmath>
#include <cstdio>
#include <vector>
#include <iostream>
#include <algorithm>
using namespace std;

static int student_id, professor_id;
class Person
{
    public:

    int age;
    string name;

    // = 0 indicates pure virtual function, must be overridden
    virtual void getdata() = 0;

    // virtual function with definition, does not have to be overridden
    virtual void putdata() = 0;
};

class Professor : public Person
{
    public:
    int publications;
    int id;

    Professor()
    {
        ++professor_id;
    }

    void getdata()
    {
        cin >> name >> age >> publications;
        id = professor_id;
    }

    void putdata()
    {
        cout << name << " " << age << " " << publications << " " << id << endl;
    }
};



class Student : public Person
{
    public:
    int marks[6];
    int marks_sum;
    int id;

    Student()
    {
       ++student_id;
    }

    void getdata()
    {
        cin >> name >> age;
        id = student_id;
        for(int i = 0; i < 6; i++)
        {
            cin >> marks[i];
            marks_sum += marks[i];
        }
    }

    void putdata()
    {
        cout << name << " " << age << " " << marks_sum << " " << id << endl;
    }
};
```

```cpp
int main(){

    int n, val;
    cin>>n; //The number of objects that is going to be created.
    Person *per[n];

    for(int i = 0;i < n;i++){

        cin>>val;
        if(val == 1){
            // If val is 1 current object is of type Professor
            per[i] = new Professor;

        }
        else per[i] = new Student; // Else the current object is of type Student

        per[i]->getdata(); // Get the data from the user.

    }

    for(int i=0;i<n;i++)
        per[i]->putdata(); // Print the required output for each object.

    return 0;

}

```

Line: 75 Col: 3

Upload Code as File □ Test against custom input Run Code Submit Code

Congrats, you solved this challenge!

✓ Test Case #0 ✓ Test Case #1 ✓ Test Case #2

✓ Test Case #3 ✓ Test Case #4 ✓ Test Case #5

✓ Test Case #6 ✓ Test Case #7 ✓ Test Case #8

✓ Test Case #9 ✓ Test Case #10 ✓ Test Case #11

✓ Test Case #12

Next Challenge

Join us on IRC at #hackerrank on freenode for hugs or bugs.