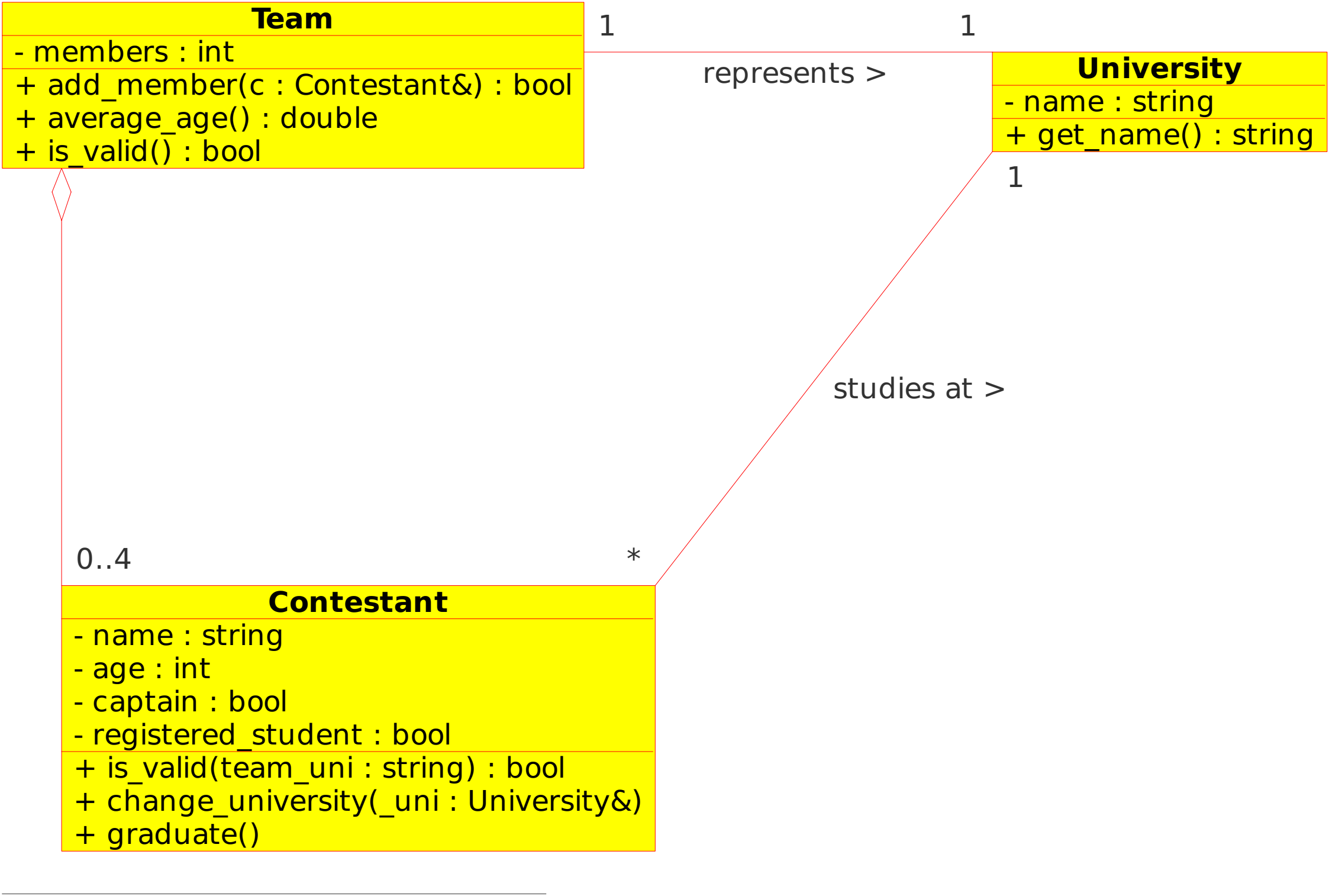
Team
- members : int
+ add_member(c : Contestant&) : bool
+ average_age() : double
+ is_valid() : bool
1
1
represents >
University
- name : string
+ get_name() : string
1
studies at >
0..4
*
Contestant
- name : string
- age : int
- captain : bool
- registered_student : bool
+ is_valid(team_uni : string) : bool
+ change_university(_uni : University&)
+ graduate()

```
#include <iostream>
#include <string>
#include <cassert>

using namespace std;

class University {
private:
  string name;
public:
  University(string _name) {
    name = _name;
  }
  string get_name() { return name; }
};

class Contestant {
private:
  string name;
  University *uni;
  int age;
  bool captain, registered_student;
public:
  Contestant(string _name, University &_uni, int _age, bool _registered_student,
 bool _captain = false) {
    name = _name;
    uni = &_uni;
    age = _age;
    captain = _captain;
    registered_student = _registered_student;
  }

  int get_age() const { return age; }
  bool get_captain() const { return captain; }
  bool is_valid(string team_uni) const {
    return (age >= 18 && age <= 40) && (registered_student) && team_uni == uni->
get_name();
  }
  void change_university(University &_uni) { uni = &_uni; }
  void graduate() { uni = NULL; registered_student = false; }
};

class Team {
private:
  Contestant *member[4];
  University *uni;
  int members;
public:
  Team(University &u) {
    members = 0;
    uni = &u;
    for (int n=0; n<4; n++)
      member[n] = NULL;
  }

  bool add_member(Contestant &c) {
    if (members >= 4)
      return false;
    member[members] = &c;
    members++;
  }
```



```
  double average_age() const {
    assert(members > 0);
    double total = 0;
    for (int n=0; n<members; n++)
      total += member[n]->get_age();
    return total/members;
  }

  bool is_valid() const {
    if (members != 4)
      return false;
    int captains = 0;
    for (int n=0; n<4; n++) {
      if (!member[n]->is_valid(uni->get_name()))
        return false;
      if (member[n]->get_captain())
        captains++;
    }
    return (captains == 1) && average_age() < 25.0;
  }
};

int main() {
  University imperial("Imperial");
  Team team(imperial);

  Contestant ivor("Ivor Bigbrain", imperial, 20, true),
    prezza("Prezza Buzza", imperial, 18, true),
    ivonna("Ivonna Singh", imperial, 25, true, true),
    yuman("Yuman Google", imperial, 32, true);

  team.add_member(ivor);
  team.add_member(prezza);
  team.add_member(ivonna);
  team.add_member(yuman);

  prezza.graduate();

  cout << "Imperial's team is ";
  if (!team.is_valid())
    cout << "NOT ";
  cout << "valid." << endl;

  return 0;
}
```