

Programming 2 – Laboratory 10 a

Your task is to implement class **Student** derived from class **Person**. Class **Person** contains dynamically allocated arrays which store a person's first and last name. Class **Student** stores dynamically allocated array of student scores. Class **Person** is based on an abstract **Individual** class.

Part 1 (2 points)

Edit the **Individual** class properties that it can be used as base class for **Person** and **Student**,

- Add a `std::string` `sign;` field for a signature of **Individual** and derived classes,
- Define pure virtual `void` `signature()` method which set a signature value,
- An `void` `info(std::ostream &)` method which takes an output stream as a parameter and format output, function should be implemented for all classes derived,
- `operator<<` which use the `void` `info()` method.

Part 2 (1 points)

In class **Person** define missing methods:

- specify method `signature()` which place initials (first letters of names) and age in `sign` field,
- define `info()` method according to the sample output given below task description,
- add other required members (if needed).

Do not define `operator<<`.

Part 3 (3 point)

In class **Student** define:

- constructor,
- copy constructor,
- `operator=`,
- destructor,
- method `double` `average()` which calculate an average score for a student.
- method `signature()` which store first 8 characters from last name, method should set value only in case the names are not empty strings;
- method `void` `Student::convert_scores(const string &in)` which set scores based on argument string, the function is used in constructor only (you can define it as last, however output may differ until it is implemented),

Do not define `operator<<`.

Constructors and operator must use the method: `void Student::init_scores(int number_of_scores, double* _scores)`

Part 4 (2 point)

Define

- function `void printlist(int n, Student* gr)` which print an array of **Students**
- function `void sortStudents()` which sort an array of **Students** . Sort function should take as a parameter a function which specify various sorting. The function return a bool value and take as parameters two **Student** objects. Make these functions a friend functions for **Students** class in order to achieve an access.

Sample output:

Output formating test

This long line| 1.2

Wheras short| 3.9

----- PART 1-2 -----

Class Person test:

os:

Mike Onesmith 20 (sign:MO 20)

os1:

Mike Onesmith 20 (sign:MO 20)

os2:

Mike Onesmith 20 (sign:MO 20)

----- PART 3 -----

Class Student test

st: Mike Onesmith

- average: 4.56 scores: [5.00 5.00 5.00 4.50 5.00 4.00 3.00 5.00] (sign:Onesmith)

st1: Mike Onesmith

- average: 4.56 scores: [5.00 5.00 5.00 4.50 5.00 4.00 3.00 5.00] (sign:Onesmith)

st2: Mike Onesmith

- average: 4.56 scores: [5.00 5.00 5.00 4.50 5.00 4.00 3.00 5.00] (sign:Onesmith)

----- PART 4 -----

GROUP - STUDENTS:

Jack One

- average: 4.56 scores: [5.00 5.00 5.00 4.50 5.00 4.00 3.00 5.00] (sign:One)

Tom Two

- average: 3.17 scores: [3.50 4.00 2.00] (sign:Two)

Adam Tree

- average: 0.00 scores: [0.00] (sign:Tree)

John Four

- average: 3.75 scores: [5.00 3.50 4.50 3.50 4.00 2.00] (sign:Four)

Nathan Five

- average: 3.30 scores: [3.50 4.00 4.00 3.00 2.00] (sign:Five)

Andrew Tree

- average: 0.00 scores: [0.00] (sign:Tree)

Jack Seven

- average: 3.19 scores: [2.00 5.00 0.00 4.00 5.00 3.50 4.00 2.00] (sign:Seven)

Mark Three

- average: 0.00 scores: [0.00] (sign:Three)

Jacky Nine
- average: 3.17 scores: [3.50 4.00 5.00 2.00 2.00 2.50] (sign:Nine)
Tom Nine
- average: 3.50 scores: [4.00 4.50 5.00 3.00 3.50 4.00 2.00 2.00] (sign:Nine)
GROUP - STUDENTS after swap:
Tom Two
- average: 3.17 scores: [3.50 4.00 2.00] (sign:Two)
Jack One
- average: 4.56 scores: [5.00 5.00 5.00 4.50 5.00 4.00 3.00 5.00] (sign:One)

Adam Tree
- average: 0.00 scores: [0.00] (sign:Tree)
John Four
- average: 3.75 scores: [5.00 3.50 4.50 3.50 4.00 2.00] (sign:Four)
Nathan Five
- average: 3.30 scores: [3.50 4.00 4.00 3.00 2.00] (sign:Five)
Andrew Tree
- average: 0.00 scores: [0.00] (sign:Tree)
Jack Seven
- average: 3.19 scores: [2.00 5.00 0.00 4.00 5.00 3.50 4.00 2.00] (sign:Seven)
Mark Three
- average: 0.00 scores: [0.00] (sign:Three)
Jacky Nine
- average: 3.17 scores: [3.50 4.00 5.00 2.00 2.00 2.50] (sign:Nine)
Tom Nine
- average: 3.50 scores: [4.00 4.50 5.00 3.00 3.50 4.00 2.00 2.00] (sign:Nine)

SORTING
GROUP - STUDENTS:
Adam Tree
- average: 0.00 scores: [0.00] (sign:Tree)
Andrew Tree
- average: 0.00 scores: [0.00] (sign:Tree)
Mark Three
- average: 0.00 scores: [0.00] (sign:Three)
Tom Two
- average: 3.17 scores: [3.50 4.00 2.00] (sign:Two)
Jacky Nine
- average: 3.17 scores: [3.50 4.00 5.00 2.00 2.00 2.50] (sign:Nine)
Jack Seven
- average: 3.19 scores: [2.00 5.00 0.00 4.00 5.00 3.50 4.00 2.00] (sign:Seven)
Nathan Five
- average: 3.30 scores: [3.50 4.00 4.00 3.00 2.00] (sign:Five)
Tom Nine
- average: 3.50 scores: [4.00 4.50 5.00 3.00 3.50 4.00 2.00 2.00] (sign:Nine)
John Four
- average: 3.75 scores: [5.00 3.50 4.50 3.50 4.00 2.00] (sign:Four)
Jack One
- average: 4.56 scores: [5.00 5.00 5.00 4.50 5.00 4.00 3.00 5.00] (sign:One)

GROUP - STUDENTS: sorted by average score
Adam Tree
- average: 0.00 scores: [0.00] (sign:Tree)

Andrew Tree
- average: 0.00 scores: [0.00] (sign:Tree)
Mark Three
- average: 0.00 scores: [0.00] (sign:Three)
Tom Two
- average: 3.17 scores: [3.50 4.00 2.00] (sign:Two)
Jacky Nine
- average: 3.17 scores: [3.50 4.00 5.00 2.00 2.00 2.50] (sign:Nine)
Jack Seven
- average: 3.19 scores: [2.00 5.00 0.00 4.00 5.00 3.50 4.00 2.00] (sign:Seven)
Nathan Five
- average: 3.30 scores: [3.50 4.00 4.00 3.00 2.00] (sign:Five)
Tom Nine
- average: 3.50 scores: [4.00 4.50 5.00 3.00 3.50 4.00 2.00 2.00] (sign:Nine)
John Four
- average: 3.75 scores: [5.00 3.50 4.50 3.50 4.00 2.00] (sign:Four)
Jack One
- average: 4.56 scores: [5.00 5.00 5.00 4.50 5.00 4.00 3.00 5.00] (sign:One)

GROUP - STUDENTS: sorted by age :

Andrew Tree
- average: 0.00 scores: [0.00] (sign:Tree)
Jack One
- average: 4.56 scores: [5.00 5.00 5.00 4.50 5.00 4.00 3.00 5.00] (sign:One)

Adam Tree
- average: 0.00 scores: [0.00] (sign:Tree)
John Four
- average: 3.75 scores: [5.00 3.50 4.50 3.50 4.00 2.00] (sign:Four)
Tom Two
- average: 3.17 scores: [3.50 4.00 2.00] (sign:Two)
Tom Nine
- average: 3.50 scores: [4.00 4.50 5.00 3.00 3.50 4.00 2.00 2.00] (sign:Nine)
Nathan Five
- average: 3.30 scores: [3.50 4.00 4.00 3.00 2.00] (sign:Five)
Mark Three
- average: 0.00 scores: [0.00] (sign:Three)
Jacky Nine
- average: 3.17 scores: [3.50 4.00 5.00 2.00 2.00 2.50] (sign:Nine)
Jack Seven
- average: 3.19 scores: [2.00 5.00 0.00 4.00 5.00 3.50 4.00 2.00] (sign:Seven)

GROUP - STUDENTS: sorted by average, last first:

Mark Three
- average: 0.00 scores: [0.00] (sign:Three)
Andrew Tree
- average: 0.00 scores: [0.00] (sign:Tree)
Adam Tree
- average: 0.00 scores: [0.00] (sign:Tree)
Jacky Nine
- average: 3.17 scores: [3.50 4.00 5.00 2.00 2.00 2.50] (sign:Nine)
Tom Two
- average: 3.17 scores: [3.50 4.00 2.00] (sign:Two)
Jack Seven
- average: 3.19 scores: [2.00 5.00 0.00 4.00 5.00 3.50 4.00 2.00] (sign:Seven)

n)

Nathan Five

- average: 3.30 scores: [3.50 4.00 4.00 3.00 2.00] (sign:Five)

Tom Nine

- average: 3.50 scores: [4.00 4.50 5.00 3.00 3.50 4.00 2.00 2.00] (sign:Nine

)

John Four

- average: 3.75 scores: [5.00 3.50 4.50 3.50 4.00 2.00] (sign:Four)

Jack One

- average: 4.56 scores: [5.00 5.00 5.00 4.50 5.00 4.00 3.00 5.00] (sign:One)