deep se
dependable evolvable pervasive software engineering group

Lab Session: Streams

Giovanni Quattrocchi

Politecnico di Milano

danilo.ardagna@polimi.it

POLITECNICO DI MILANO

The Problem

Goal: to provide a program that reads from different files information about students and their exams and stores them in a suitable data structure.

Students are characterized by name, surname, birth date and a `vector` of exams. Moreover, students are uniquely identified by an ID.

Exams are represented by the ID of the course, the date and the grade.

The correspondence between students and exams is expressed through the students' ID.

The Problem

The list of students is provided in file `students.txt`, which stores in each line the student's ID, his name, surname and birth date. Fields are separated by commas, i.e. `students.txt` is a Comma Separated Value (CSV) file.

Example:

```
100100,Mario,Rossi,8/6/1991
```

The list of exams is provided in file `exams.txt`, which stores in each line the student's ID, the course's ID, the date and the grade. Fields are separated by commas.

Example:

```
100100,101,8/5/2019,28
```

The Problem

Data coming from `students.txt` and `exams.txt` must be stored in a

`unordered_map<unsigned,StudentsData>`

denoted by the user-defined type

`students_type`

The keys of the map are given by the students' IDs.

The following definitions are given

```
typedef unordered_map<unsigned,StudentsData> students_type;
typedef vector<string> row_type;
typedef vector<row_type> table_type;
```

The Problem

You have to implement:

1) A class `FileManager` that can be used to read a generic CSV file and that stores the corresponding elements in

```
table type fields;
```

In particular, files are read through the method

```
const table_type& parse_file (const std::string& filename,
                              char d = ',');
```

It receives as parameters the name of the file to be read and the char that is used in the file to separate the different fields (its default value is comma). It returns the table `fields` storing the values read from file (as a vector of vector of strings!).

The Problem

2) Two functions

```
void add_students (const table_type&, students_type&);
```

and

```
void add_exams (const table_type&, students_type&);
```

that receive as parameters the table `fields` created by `FileManager` and the map of students' data and add to the map the students read from `students.txt` and the exams read from `exams.txt`, respectively.

Note: it is not possible to add exams to a student if the student is not stored in the map!

3) Print, for each student the average of their grades

REMARK

Remember to import the initial code in CLion as an **existing project**, instead of directly opening it, in order to be able to build and execute the given code!