POO (09:00-10:00)

Write the class **TextAnalyzer** so that the following code

```
#include "TextAnalyzer.h"

int main()
{
    try {
        TextAnalyzer t("phrase.txt");
        std::cout << "Sentences: " << t.sentences_count() << std::endl;
        for (int tr = 0; tr < t.sentences_count(); tr++) {
            std::cout << " * " << t.sentence(tr) << std::endl;
        }
        std::cout << "Words : " << t.words_count() << std::endl;
        for (int tr = 0; tr < t.words_count(); tr++) {
            auto [w, f] = t.word(tr);
            std::cout << " * " << w << " -> frequency: " << f << std::endl;
        }
        for (auto ch = 'a'; ch <= 'z'; ch++) {
            if (t.frequency(ch)>0)
                std::cout << ch << "=" << t.frequency(ch) << ", ";
        }
        std::cout << std::endl;
        try {
            auto s = t.sentence(100);
        }
        catch (std::exception e) {
            std::cout << "Error: " << e.what() << std::endl;
        }
        try {
            auto w = t.word(100);
        }
        catch (std::exception e) {
            std::cout << "Error: " << e.what() << std::endl;
        }
    }
    catch (std::exception e) {
        std::cout << "Fail to open file !";
    }
}
```

compiles and upon execution prints the following to the screen:

```
Sentences: 3
 * Ana are mere si pere
 *  Ana are mere
 *  Ana nu are gutui
Words : 7
 * Ana -> frequency: 3
 * are -> frequency: 3
 * mere -> frequency: 2
 * si -> frequency: 1
 * pere -> frequency: 1
 * nu -> frequency: 1
 * gutui -> frequency: 1
a=9, e=9, g=1, i=2, m=2, n=4, p=1, r=6, s=1, t=1, u=3,
Error: Invalid index for sentence !
Error: Invalid index for word !
```

Carefully read the main function to deduce what methods/operators should be included in CSVLoader class.

Observations:

- For the previous example, the “phrase.txt” is located in your archive
- Use STL to solve this problem
- Use a map for character frequencies
- The code should be compatible with C++17 or beyond

Grading (informative):

G1	Organize your project in 3 files: **main.cpp**, **TextAnalyzer.h** and **TextAnalyzer.cpp**	2p
G2	Constructor that reads the text file and process it	8p
G3	Exception support in constructor	2p
G4	File is being open and read via std::ifstream	2p
G5	Method sentences_count	2p
G6	Method words_count	2p
G7	Method sentence	2p
G8	Exception for method sentence	2p
G9	Method word that returns the value a tuple with the word and its frequency	5p
G10	Exception for method word	2p
G11	Method frequency (that uses a map to query the frequency !)	1p