

## **TP1 – Getting started with C++ Data Structures for Meshes**

### **Open and run a C++ example file**

Open the file `LaClasseCentrale.cpp` and try to predict its behavior.

Compilation with the command

```
g++ -std=c++17 -Wall LaClassePourSeMettreEnRoute.cpp  
-o vasy
```

Run the execution file `./vasy` and understand the differences with what you expected.

### **Create a mesh data-structure in C++**

- 1)** Write the code of the triangulated mesh data structure with geometry and connectivity information.
  - First, you model a mesh by a vector of its vertices and a vector of its faces. In this first model, the faces are not attached together. They are simply represented by the indexes of their 3 vertices.
  - Then you write the data structure corresponding to the topological model based on vertices and faces that we saw in class. In this model, the faces are attached together.
- 2)** Construction of elementary meshes to test your data structures.
  - A tetrahedron (be careful when attaching the faces together)
  - A pyramid with a square base
  - A 2D bounding box (composed of 2 triangles) whose edges are connected to an artificial “infinite” vertex at the back.
- 3)** Write a routine to save a mesh in a file, using an OFF format :  
OFF  
Number of vertices s  
Number of faces c  
Description of the faces (sequence of the indices of the vertices of the face, preceded by its number of vertices).
- 4)** Read and load, in your mesh data structure, a triangulated mesh written in an OFF format. This is not straightforward since the

connectivity between the faces is not provided by the OFF format.

Ensure that you implement a careful software approach.

### **Open and visualize a mesh stored in a off file**

You can use an existing viewer :

- Install the 3D model viewer meshlab.
- Visualize the mesh queen.off.

You can even find online viewer such as 3dviewer.net

You can develop your own viewer with an associated GUI using Qt.