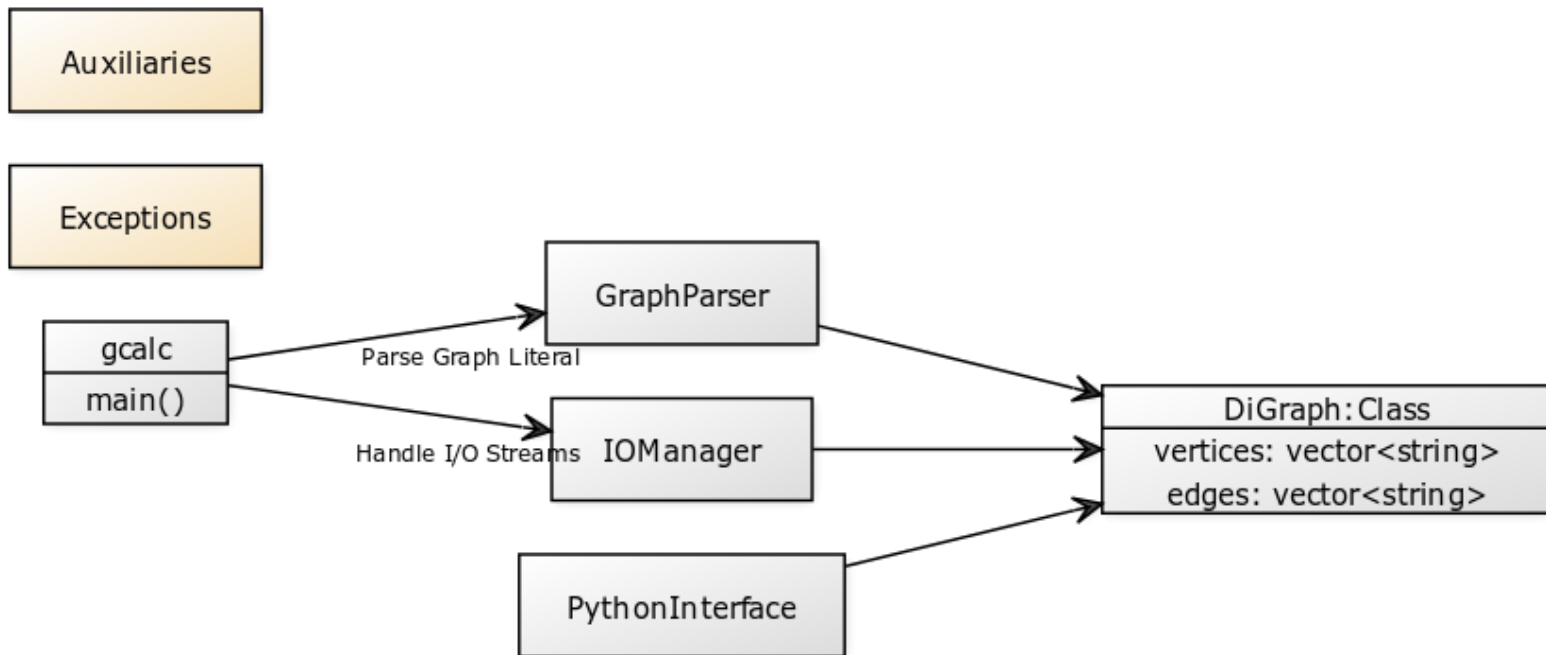


Final project design



Design abstract

Core Mechanism (Gcalc.h/cpp) & Parsing Logic (GraphParser.h/cpp)

The **gcalc** program runs a loop prompting the user to insert known commands to the program interactively. If gcalc runs as batch mode, then the standard input/output stream are replaced by the given files.

GraphParser Parses a string literal according to precedence rules in a recursively manner with the given string(payload) in a function argument or as a right-hand assignment literal

The parser delaminates a given string to graph operands and runs the parser recursively on each operand literal.

$$\text{Parse}(\underbrace{G1}_{\text{Parse}(G1)} + \underbrace{! G2}_{\text{Parse}(! G2)} ^ \underbrace{(\{...\} - G4)}_{\text{Parse}(\{...\} - G4)} * \underbrace{\text{load}(\text{file.txt})}_{\text{load}(\text{file.txt})})$$

The parser relies on string manipulation functions, and thus includes **Auxiliaries.h**, and requires loading graph files, therefore includes the **IOManager.h**.

Helper files (IOManager, Auxiliares cpp/h)

IOManager provides tools for interfacing with files, whether in batch mode or regular graph files.

Auxiliaries provide tools for manipulating strings, such as split by delimiters, parentheses balancing, and string trimming of white spaces.

Python Interface (PythonInterface.cpp/h)

PythonInterface defines interface functions that enables scenarios and operation to be executed by an external python interpreter.

Classes:

DiGraph Class: The DiGraph class represents a directed graph object that consists of vertices and edges of type string and enables some methods to be used externally.

All project objects include the **Exceptions.h** for handling exceptions.