Students Grade Calculator

This C++ application allows users to enter, calculate, and manage student grades efficiently. It uses modern object-oriented programming techniques, including the Rule of Three, and supports dynamic homework handling with `std::vector`. The program can calculate final grades using either the average or the median and can read student data from a file or generate it randomly. The output is clean, formatted, and sorted by student names.

Key Features

- Dynamic homework input using `std::vector`
- Final grade calculation using **average or median** (user's choice)
- Random generation of homework and exam scores
- Importing student data from external files
- Clean, well-formatted output sorted by name or surname
- Full Rule of Three implementation in the `Person` class
- Extensible and easy-to-maintain codebase
- Final Grade Formula:
  Final = 0.4 × (HW Average/Median) + 0.6 × Exam

Release V0.1

Created a class `Person` storing:

- First name and surname
- Homework results
- Exam result
- Final grade

Implemented:

- Constructors, destructor, copy constructor, copy assignment operator (Rule of Three)
- Overloaded input/output operators (`cin` and `cout`)

Name Surname Final (Avg.) | Final (Med.)

Name1 Surname1 x.xx | y.yy

Release V0.2

Improved structure and performance by:

- Splitting the program into multiple `.cpp` and `.h` files
- Adding exception handling for file operations and container access
- Enhancing random data generation (10K–10M records)
- Automatically sorting students and dividing them into “passed” and “failed”
- Saving sorted results into separate files
- Measuring program speed using `std::chrono`

Release V0.25

- Performed detailed speed testing on datasets of various sizes
- Created versions using std::list and std::deque instead of only std::vector
- Compared two splitting strategies:
    - Copying students into two new containers
    - Moving/removing students directly from the original container
- Analysed memory usage and performance for each method

Release v1.0

- Optimized splitting and sorting using STL algorithms: `std::copy_if`, `std::remove_if`, `std::back_inserter`
- Mainly used `std::list` for its efficient insert/erase operations
- Achieved significant performance improvements and cleaner code

How to Use the Application

1. Preparation
- Ensure all source files (**main.cpp**, **Person.cpp**, **Person.h**, **Lib.h**) are in one folder
- Install a C++ compiler (g++, clang, or MSVC)
2. Build with Make (Linux/Mac)

make

3. Build with CMake (Windows/Linux/Mac)
cmake .
cmake --build

4. Run the Program
./main

The program will:

- Generate several random student data files
- Read one file, sort students, split into “passed” and “failed”
- Save results to output files
- Show execution times for reading, sorting, splitting, and writing
- Report any errors (e.g., missing files)