# 06. Exercises on Advanced Members & Code Organization

Write C++ code for solving the tasks on the following pages.

Code should compile under the C++03 or the C++11 standard.

Submit your solutions here: <https://judge.softuni.bg/Contests/1253/06-Exercises-on-Advanced-Class-Members-and-Code-Organization> (select “Compete” when prompted)

Any code files that are part of the task are provided under the folder **Skeleton**.

Please follow the exact instructions on uploading the solutions for each task.

# Task 2 – Notes

You are given code for a program that reads musical note names in the solfège naming convention (Do–Re–Mi–Fa–Sol–La–Si) and translates them into NoteName objects following the English naming convention (C–D–E–F–G–A–B), however the provided code is missing the translation logic.

Your task is to study the way the provided code uses the translation logic and implement the translation logic so that the code compiles successfully and accomplishes the task described.

You should submit a single .zip file for this task, containing ONLY the files you created.

The Judge system has a copy of the other files and will compile them, along with your file(s), in the same directory.

### Restrictions & Hints

You can assume the code works correctly and the input will always lead to a valid sequence of commands. Focus on the hint-printing part, you can mostly ignore the rest (*though it would be a good idea to study the code to see how such programs may be implemented*).

In addition to the skeleton, you are given a Windows executable that behaves correctly so you can study the wanted behavior before implementing the necessary code.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| Do Re Mi Fa Sol La Si unknown end | C D E F G A B ? |