# 09. C++ OOP Pure Virtual Members and Multiple Inheritance

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/1281/09-Cpp-OOP-Pure-Virtual-Members-and-Multiple-Inheritance> (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.

NOTE: the Judge system treats each .cpp file as a compilation unit, compiles each such file and links them together to create the final executable, which is checked against the tests.

# Task 4 – Tree

Like in **Task 3**, you are given code which reads information File and Directory objects in a file system, each of which has an id, and each of which has a parent – the object which contains it.

Operations with the files and directories are done with reference to their **id**, and there are several types of operations:

* file – create a File object with a filename and **contents** (a sequence of characters, stored in a string)
* directory– create a Directory object with a name
* move– move an object (File or Directory) into a Directory
* shortcut– creates a “shortcut” to a file or directory. Shortcuts do not move the object (i.e. the object remains in the directory it was originally, but it also appears in the shortcuts)

The provided code is missing the definitions for the File and Directory classes – you should implement them.

Note: there are some minor changes to the requirements for file system objects (getters and range-based for loop usability for FileSystemObjectsContainer).

Your task (in addition to implementing File and Directory) is to implement a “tree view” for the file system entered on the input.

A tree view is a layered representation of hierarchical objects. Objects on the **first level are printed without indentation**. Objects on the **second level** (i.e. objects contained inside directories from the first level) are printed with **1 level of indentation** after their **parent** objects. And so on – **objects on each following level are printed with an additional level of indentation, compared to their parents**, and are printed **on the line after their parents**. Additionally, objects on **each level** should be **sorted lexicographically** (using operator< of the string class). The shortcuts are printed as if they were a directory named [shortcuts].

For this task, one level of indentation should be represented by the string "--->" (three dashes and a “greater than” sign, i.e. an arrow). See the examples below for more details on how to represent the tree view.

You should submit a single .zip file for this task, containing ONLY the file(s) YOU created. The Judge system has a copy of the other files and will compile them, along with your file, in the same directory.

### Restrictions

The input will always contain correct operations – i.e. any object used by an operation will have already been created by the file or directory operations. There will be no invalid or duplicate ids, no move/shortcut operations referencing ids not yet created. No object (File or Directory) will have the same name as another object.

The provided code handles input/output and operation management – you should focus on implementing the classes it uses and on implementing the construction of the tree view.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| file 1 example.txt some example bytes as text  file 2 otherFile.txt other text  directory 3 examples  move 1 3  directory 4 nested  move 4 3  move 2 4  file 5 rootFile.txt this file is in the file system root  directory 6 rootDir  end | examples  --->example.txt  --->nested  --->--->otherFile.txt  rootDir  rootFile.txt |
| file 1 example.txt some example bytes as text  file 2 otherFile.txt other text  shortcut 2  directory 3 examples  move 1 3  directory 4 nested  shortcut 4  move 4 3  move 2 4  file 5 rootFile.txt this file is in the file system root  directory 6 rootDir  file 7 noDot can't use name to check if directory or file :)  move 7 4  shortcut 6  end | [shortcuts]  --->nested  --->--->noDot  --->--->otherFile.txt  --->otherFile.txt  --->rootDir  examples  --->example.txt  --->nested  --->--->noDot  --->--->otherFile.txt  rootDir  rootFile.txt |