# 07. Destructors, Constructors and Copy-Assignment

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/1250/07-Destructors-Constructors-and-Copy-Assignment> (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 3 – Array of Pointers

**NOTE**: this task is the same as **Task 1 – Register**, and **Task 2 – Register of Three,** however it does NOT use a Registerclass, but instead allocates Company objects in dynamic memory and uses an ArrayOfPointers class to store these pointers (NOTE: the provided code does NOT deallocate the memory it allocates).

You are given code which reads information about Company objects from the console, parses it multiple times (the number of repetitions is entered on the first line on the console) and prints the information about one of the Company objects, specified by its **id**.

The provided code handles input, output, and the repeated executions – your task is to **declare** and **implement** the ArrayOfPointers class in any way you think will accomplish the task, **without leaking memory**. The provided code expects the declaration to be in an ArrayOfPointers.h file, but you are free to chose whether to implement the class in the same file, in a .cpp file, or in multiple files.

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

There will always be a Company with the specified **id**. Make sure there are no memory leaks.

### Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1  2  42 theanswer  69 thehub  42  end | 42 theanswer |
| 1000  2  42 theanswer  69 thehub  42  end | 42 theanswer |