IP1

Requirements:

- provide code for a single class in .cpp file subject to the requirements listed below

The class should have:

- at least three fields: at least one is a string and at least one number
- setters and getters for every field
- at least two constructors (no default one) and a destructor
- setting at least one field should require validation, throw an exception if it does not pass
- toString method to convert an object to text (use stringstream for this)
- automatic numeration of all objects (give a unique id to each instance)
- automatic counting objects in memory (have counter used in constructors/destructor)
- proper layout (fields, constructors/destructors, then other methods)
- proper style (naming, use of whitespace, etc)

The main function should have:

- unit tests for the class (use assert or other means to test)
- no user actions required
- change every field and call every method at least once
- create at least one object dynamically (new and delete)
- create a list of objects for the class
- test1: create an object getters return the provided values and toString returns what it must
- test2: for every public setter call it and check whether the old value properly changes into a new one
- test3: check if failed validation throws and exception of the desired type
- test4: check whether automatic numbering works properly (id can only increase)
- test5: create a list of objects for the class dynamically (using new and later delete), and make sure the count of objects in memory is valid (0 in the beginning, increases with new and decreases with delete, and 0 in the end)
- also ensure that all exceptions get caught and no memory leak occurred

Timeframe:

- to be submitted during February.