Computer Lab #6.

Constants and Static Class Members in C++

In general (and in the C++ programming language in particular), the String class is a standard representation for a string of text. This class removes many of the problems introduced by C-style strings by putting the responsibility of memory management on the String class rather than on the programmer. The class should provide some typical string operations like comparison, concatenation, random access to elements, length, etc.
The aim of the Computer Lab is to practice dynamic class attributes, constants, const reference parameters, and static class members as well as development of general programming skills in C++.

Task 1.

Download the source code from the Course Webpage. Open the downloaded archive and add ***String.h*** and ***StringTest.cpp*** files to your C++ project. Create the missing ***String.cpp*** source file and add it to the project. Follow comments and instructions in the downloaded code.

a) Complete the ***String.h*** header file by writing the missing functions declarations.
b) Insert the missing *const* keywords at parameter passing wherever it makes sense. Remember that *const* has no effect on parameters passed by value.
c) Substitute function parameters to those passed by constant reference wherever it is worth doing so. Remember that it is not worth passing built-in types by const reference.
d) Write functions implementations in ***String.cpp***. Pay attention to namespaces. Implementation for *print(...)* function is given below.
e) Start the program in debugger mode and observe its execution in ***StringTest.cpp***. When will a copy constructor and a destructor be called? Why?
f) Show that if a static function is substituted by a non-static one, then the latter refers to a class object, and not to the class itself. For example:

```
static bool compare(const String& string1, const String& string2);
// compare: string1 to string2
bool compare(const String& string1);
// compare: this to string1
```

Implementation of *print(...)* function:

```
void String::print(ostream& os)
{
   for(unsigned int i=0;i<elementsNum;i++)
   {
      os<<pData[i];
   }
}
```

Decomposition of the program:

- *String.h* file: class declaration header;
- *String.cpp*: implementation of class members declared in the header;
- *StringTest.cpp*: the main program with testing code.