Software Design (C++), 2014

Course project report

`Aaro Lehikoinen`
`15.12.2014`

String class provides easy and safe string operations, which makes programming much easier, since almost all programs use strings. This C++ class String is a simple and safe implementation for basic small string operations. It's not performance optimized, but works well with small strings.

Strings are saved as an array of `char`s. This internal data structure is allocated more memory if some operation adds chars to the string. All operations allocate only required amount of memory. To optimize for longer strings and heavy usage of concatenation, this behavior could be changed. The amount of allocated memory is never reduced, only more is allocated.

Class invariant of String states that the length of the stored string should be less or equals the size of allocaten memory and if the size of allocated memory is 0, the pointer to the data should be a null pointer.

The class has one helper function `memcopy()` which copies a block of memory elsewhere. This function can be used also with overlapping memory segments, it detects them and copies the memory safely. Before realizing this was necessary, it caused some confusion.

Basic operators are overloaded and a string literal constructor is implemented. Simple iterators exists. Stream operators `<<` and `>>` are implemented. `<<` operator reads the stream until the stream is closed (EOF or error).

All functions check for pre-conditions and throws an exception if pre-conditions are not met. Post-conditions are checked after every public funtion that should leave the object to a consistent state.

Usage

This program has been coded and tested in a Linux environment and uses GCC C++11 compiler. Attached Makefiles should provide all necessary commands to compile, test and execute the program.

Build the program by executing command `make build`. A “production” version with asserts disables can be build with `make build-prod`. Build binaries can be found from `bin` directory.

To run build and run tests type `make test` to terminal. This prints all the test descriptions, it's passing status and possible errors.