Advanced C++ Programming

Assignment Sheet 2

Preliminaries

In the following, you will be working with 4 files. The `CMakeFile` states how the assignment is compiled. The main function and the tests are in the `main.cpp`. The `Vector` class, which you will have to complete has a header file `Vector.h` and a cpp file `Vector.cpp`. The completion of the `Vector` class is done only in the `Vector.cpp`[1]. This is where your solution will be, which you upload once you are finished. Your solution has to work with the provided (and unchanged) `CMakeFile`, `main.cpp` and `Vector.h` files. If you are unable to compile, run and valgrind your code, you can use our online validator. When you are done, upload your solution (named as `Vector.cpp`!) to StudOn.

1 Vector Class

Implement a vector class in `Vector.cpp`. Your class has to pass the provided tests and it must not have any memory leaks.

In the first assignment we have implemented a Vector class with our own memory allocation. This can be achieved with allocators more easily and flexibly. Allocators serve as an abstraction to translate the need to use memory into a raw call for memory[2]. We will use the standard allocator[3]. Your task is to implement the Vector class utilizing the `std::allocator`.

- The project should compile as provided and execution will result into an error.
- The first testcase `test_get_set` should be working from the get go.
- You can use the `print` function in main for debugging purposes.
- Have a look at the `Vector.h` file. You will find the functions, which need to be implemented. Notice that some function and the private class members are given.
- Your Vector already contains the function `size()`, which returns the size of the Vector, `begin()` and `end()`, which can be used to iterate over the `elements` pointer and a `capacity` function, which retruns the current capacity of the Vector.
- Implement the `free`, `reallocate` and `push_back` functions.
- Implement all constructors, the destructor and satisfy the "rule of 5". Use the `free` and `alloc_n_copy` where appropriate.

  Hint: The initializer list assignment is already implemented and can.

[1] We use the extension ".cpp", which is not a proper extension. Since `Vector` is a templated class, the definition of it needs to be contained in the header file. However we want to have it in a different file. In order not to confuse some IDEs, we are not using the ".cpp" extension and arbitrarily chosen ".cpp". At the end of the header file, there is an include if the "template" file, which makes the deffinition of Vector's functions available.

[2] Codeguru-Article

[3] `std::allocator`

- Make sure your implementation works for all kinds of templates.
- Compile and run your project with the provided `CMakeFile`.

Make sure your implementation passes the provided tests! Run your program also with `-fsanitize=address` and with `valgrind`[4] to detect memory leaks or invalid accesses. Hint: In some environments `valgrind` may report false positives. You may want to check this by comparing the `valgrind` output of your program to the `valgrind` output of a minimal program containing only an empty main function.

[4] https://valgrind.org/