EECS 560 Lab 0: Single Number

Objective

- Get familiar with coding with C++ under the Linux environment (Ubuntu).
- Get familiar with ADT implementation with C++ and the lab setup of this course.
- Recap C++ fundamentals such as object, constructor and destructor, template, and overloading.

Specification of the data structure

- Define your data structure as "`template <typename DataType> class MyNumber`".
- Constructor functions:
  - `explicit MyNumber(DataType rhs = 0)` // default constructor
  - `MyNumber(const MyNumber<DataType> & rhs)` // copy constructor
  - `MyNumber(MyNumber<DataType> && rhs)` // move constructor
- Destructor function:
  - `~MyNumber(void)` // collects allocated memory before the destruction of the object
- Data access:
  - `DataType read(void) const` // returns the number being stored
- Data update:
  - `void write(DataType rhs)` // record the parameter
  - `MyNumber & operator= (const MyNumber<DataType> &rhs)` // copy assignment
  - `MyNumber & operator= (MyNumber<DataType> &&rhs)` // move assignment

Additional Requirements:

1. Install "valgrind" to your Ubuntu Linux system with the command "sudo apt install valgrind". You will need the sudo password to your Ubuntu system.
2. You should use "new" to allocate memory space to the variable that stores the data (a number with DataType).
3. You should use "delete" to collect the allocated memory space upon the destruction of the object instance.

Testing and Grading

We will test your implementation using the tester main function posted online. The posted input and output examples should be used for a testing purpose, while we will use another set of inputs for grading. Your code will be compiled under Ubuntu 20.04 LTS using g++ version 9.3.0 (default) with C++11 standard.

Your final score will be the determined by the success percentage of your program when fed with many random inputs. Note that if your code does not compile (together with our tester main function), you will receive 0. Therefore, it is very important that you ensure your implementation can be successfully compiled before submission.

Submission and Deadline

Please submit your implementation as a single .h file, with a file name "MyNumber_[YourKUID].h". For example, if my KU ID is c123z456, my submission will be a single file named "MyNumber_c124z456.h". Submissions that do not comply with the naming specification will not be graded. All submission will go through Canvas. The deadline is Friday January 27th, 2023, 11:59PM.