MEDGAR EVERS COLLEGE
The City University of New York

# Programming Project A

## Directions:

You will create a new data type (a class) named *integer* that can represent any real number integer (an integer of any length) by using a linked list. The data type must perform integer addition and subtraction. The data type must be able to be typecasted from a string, int and double. Last, it must be able to be written to/read from the terminal or file.

The class must have the following components; however, it may contain any additional fields or methods that you deem necessary.

- ☐ A private linked list field that is used to represent the integer which we will be refer to as the integer field.
- ☐ A public default constructor that initializes the integer field to 0.
- ☐ A public overloaded constructor that takes a string parameter. It converts the parameter and assigns it to the integer field if parameter is in the proper real number integer format. Otherwise, it initialize the integer field to 0.
- ☐ A public overloaded constructor that takes an int parameter. It converts the parameter and assigns it to the integer field.
- ☐ A public overloaded constructor that takes an double parameter. It converts the parameter and assigns it to the integer field. When converting a double to an int, the decimal portion is truncated.
- ☐ A public copy constructor.
- ☐ A public assignment operator.
- ☐ A public destructor. It deallocates the integer field.
- ☐ A friend overloaded addition operator that takes two constant integer reference parameters. It returns an integer object that is equal to the sum of the two parameters.
- ☐ A friend overloaded subtraction operator that takes two constant integer reference parameters. It returns an integer object that is equal to the difference of the first parameter minus the second parameter.
- ☐ An overloaded compound addition operator that takes a constant intger reference parameter. It adds the value of the parameter to the integer field.
- ☐ An overloaded compound substraction operator that takes a constant intger reference parameter. It subtracts the value of the parameter to the integer field.
- ☐ A friend overloaded equal operator that takes two constant integer reference parameters. It returns true if the values of the parameters are equal; otherwise, it returns false.
- ☐ A friend overloaded not equal operator that takes two constant integer reference parameters. It returns true if the values of the parameters are not equal; otherwise, it returns false.
- ☐ A friend overloaded greater than operator that takes two constant integer reference parameters. It returns true if the value of the first parameter is greater than the value of the second parameter; otherwise, it returns false.
- ☐ A friend overloaded less than operator that takes two constant integer reference parameters. It returns true if the value of the first parameter is less than the value of the second parameter; otherwise, it returns false.
- ☐ A friend overloaded greater than or equal to operator that takes two constant integer reference parameters. It returns true if the value of the first parameter is greater than or equal to the value of the second parameter; otherwise, it returns false.
- ☐ A friend overloaded less than or equal to operator that takes two constant integer reference parameters. It returns true if the value of the first parameter is less than or equal to the value of the second parameter; otherwise, it returns false.
- ☐ A friend overloaded ostream operator. It displays the integer field.
- ☐ A friend overloaded istream operator. It reads into the integer field.

You must adhere to the following restrictions. **Failure to follow the restrictions will result in a 0 for the project**.

- You must use class lecture defined data structures and functions, and/or define your own.
- The use of STL library classes are prohibited.
- The class must be defined in a header file.
- You must write a main cpp file to test the integer object methods.

## Grading Rubic:

Your grade will be based on the following rubric:

### Final Project Grading Rubric

| Category | Task | Points |
|---|---|---|
| **Specification** | ∘ Program compiles.<br>∘ Program performs required tasks.<br>∘ Program produces accurate and formatted outputs. | 40 |
| **Readability** | ∘ Program uses meanful identifiers.<br>∘ Program indents scopes. | 10 |
| **Documentation** | ∘ Program provides a header.<br>∘ Program provides descriptions for functions. | 10 |
| | | 60 |