

CS 211, Winter 2004  
Lab Assignment L3: Handin Instructions  
Assigned: Monday, Feb. 9, Due: Monday, Feb. 23, 11:00 AM

Brian M.Dennis, Instructor  
Bin Lin, Tom Lechner, Rachel Goldsborough, Teaching Assistants

## Overall

Tom will be the coordinator for lab 3. You will mail your submission to him at: t-lechner@northwestern.edu.

If you completed your assignment in Visual C++, you should **clean your workspace**. This means removing built executables and object files. Then you should create a zip archive of the workspace. Utilities like WinZip

<http://www.winzip.com/>

and FreeZip

<http://members.ozemail.com.au/~nulifetv/freezip/>

make this easy to do. Then e-mail your archive as an attachment named **cs211-lab3-yournetid.zip**.

**Did I mention that you should clean your workspace!!** The resulting zip archive shouldn't be more than a few KB. Definitely not over 1 MB.

If you completed your assignment on UNIX or Cygwin, try to get all of your source files into one directory. You should provide a **make** file that can build each of the various exercises. Delete any built executables and object files. You should then create a compressed tar file of the directory like so:

```
tar zcvf cs211-lab3-yournetid.tar.gz lab-dir
```

Then e-mail your archive as an attachment named **cs211-lab3-yournetid.tar.gz**

Also, in your e-mail include your name.

For this assignment there are no additional questions being asked outside of the requested classes.

Please follow the naming conventions for member functions specified in the homework and in this handout. Both for the actual code and for the archive name. It helps us make sure you get the right grade.

The same collaboration guidelines as the previous two labs applies.

## Complex

For this problem, you mainly need to make sure you've implemented all the class members that have been asked for. The driver program doesn't actually use doubles to initialize its `Complex` instances so you may

want to add tests for that yourself.

- Make sure your input operator `>>` does something reasonable with malformed input e.g. `(1.1 1.14159]`
- If you don't know what the problem is referring to when it speaks of "default values" read section 3.18. Test your `Complex` with 0, 1 and 2 arguments.
- Your `Complex` class declaration should be in C++ header file, probably best named `Complex.h`. Your implementation should be in a C++ source file similarly named `Complex.cpp`. The same naming scheme should apply for the remaining problems.

## Rational Numbers

Make sure you deal with the following issues:

- All `Rational` instances are in reduced form.
- Division by zero is detected and avoided. Returning 0 for the result is fine.

## Integer Sets

Make sure you deal with the following issues:

- That instances of your `IntegerSet` can actually hold every integer from 0 to 255.
- You may want some internal/private member functions that deal with accessing the array by index. Avoid cutting and pasting code all over the place. My rule of thumb is that if you cut and paste some code 3 times, it's time to turn it into a procedure or member function.
- That you can detect the empty set and print it out properly.
- Sets for which `isEqualTo` returns true are actually equal
- Sets for which `isEqualTo` returns false are actually different

## HugeInteger

For this assignment, you can assume all `HugeIntegers` are positive. If you have division by zero, or if subtraction results in a negative number you can simply return zero.

This one is pretty tricky. Slow and working is better than fast and broken. Also, you can recycle member functions to implement member functions. For example, you may need subtraction to implement division.

Besides what's requested here are things to look out for:

- You should probably overload the copy constructor for `HugeIntegers` to limit sharing of the integer arrays between instances
- Ditto for the assignment operator.
- Since you have to return a new `HugeInteger` for each calculation, there's really not much of a resizing issue
- Since `HugeIntegers` have dynamically allocated memory within them, you should write a destructor that cleans up that space when an instance is deallocated
- As there are a couple of member functions that are a bit complex, comments for this class will be particularly important. For modulus, multiplication, and division please document how your member function operates mathematically.
- Try your code on a couple of 50 digit integers. Using the provided driver code it should be easy to specify this from the command line.