**Lab 4 assessment – next week (internal and external students, in-person)**

There are 3 parts to be looked at.  
  
**UML** - the design. Use staruml. Low level design as a minimum.  
Note some students would look at the doxygen output diagram and then copy that design into staruml. This is the wrong approach and invariably they will have the wrong design, and thus low marks. All of these have been covered in the workshop recordings. There was a bit of discussion about it in yesterday's workshop/clinic  
  
**Code** (There was discussion at yesterday's workshop/clinic)

* No friend declaration
* operators << and >> are moved out of the class.
* All **getters** must be declared **const**. No exceptions to this, ever.
* Separation of interface from implementation
  + No inline methods (which coding rule is used here?)
* All data is private
* No global data, const globals are fine
* No unnecessary #include in .h files.
* C string is not used. Changed over to std::string which is much nicer the C string
* In main(), the following lines of code must not change. Remain exactly as is.

int main()  
{  
  ifstream infile( "rinput.txt" );   
  if( !infile ) return -1;  
  
  Registration R;    
  
  infile >> R;     
  
  ofstream ofile( "routput.txt" );  
  
  ofile << R        // also a subroutine call. Can you guess what it means?  
    << "Number of courses = " << R.GetCount() << '\n'  
    << "Total credits     = " << R.GetCredits() << '\n';  
      
// After the above, the next few lines of code would change or be commented out  
// as indicated in Lab 3 Registration  
  
   return 0;  
}

Output format needs to be the same specified in Lab 4. Note that month string is output, but the Date class is **never** to be associated with string. No code for handling string is to be found in Date.h or Date.cpp. See workshop/clinic recordings from weeks 1, 2 and last night.  
  
Also see advice about unsigned usage in week 1 and 2 recordings. Don't use unsigned.  
  
Do not return by pointer as that will expose class internals to the client, defeating the requirement to have data private to the class. Also discussed during workshop/clinic yesterday.   
  
Code must build and run.  
  
All the above is mandatory. What that means is there are no marks allocated for code if any item listed above is violated.  
  
  
**Doxygen**  
    .h files have doxygen comments using the style in Modelfile.h. Copy/paste from there and change as needed.  
  
html output shows class diagrams.   
  
 **QandA**  
Go through the lab 4 QandA. Do **not** ignore any advice here.