UML Diagrams

`StarUML is found in LMS`

1 UML

Some points to remember when doing your UML:

- Each class in your system must be included in the one UML diagram, and must have a class name provided.
  - The client program (i.e. the .cpp file with the main function in it) **DOES NOT** get included in the UML diagram. Client programs (and there may be many clients using your class system) are those programs that use possibly many class systems to help them perform certain tasks (think of how many client programs use the string or data classes). A UML diagram demonstrates only the class system you are developing; it provides information client programs need to know about each class in your class system, and the relationship between the classes.
- Each class in your system must indicate its member variables in the top part of the class rectangle.
  - Each member variable of a class **MUST** be shown as either `public, private, or protected`. In most cases in this unit, member variables will be `private` and are indicated by a minus sign.
  - Member variables are entered in the UML in the following format: `-variableName: data_type`.
- Each class in your system must indicate its member functions in the lower part of the class rectangle.
  - Each member function **MUST** be shown as either `public, private, or protected`. In most cases in this unit, member functions will be `public` and are indicated by a plus sign.
  - Member functions are entered in the UML in the following format: `+functionName(parameter_list): return_data_type`.
  - Each parameter in the parameter_list is comma separated and should follow the format for member variables. Any constant and reference parameters must also be indicated: `parameterName: const data_type &`.
  - If a function is a constant function this needs to be included in the UML, and is typically entered after the function but before the return_data_type. `+functionName(parameter_list): const: return_data_type`.

- The relationship between classes must be indicated. The particular symbols to represent the type of relationship are discussed in the lecture slides. You need to familiarize yourself with them.
  - Along with the type of relationship, you also indicate the multiplicity.

Below is included a UML for the registration class system provided to you for lab 2.

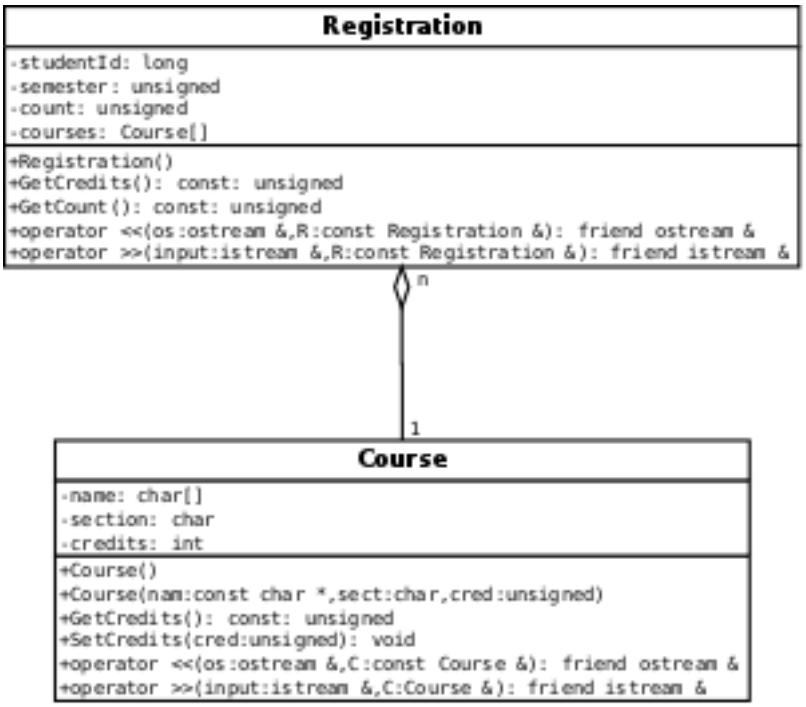

Figure 1: Registration Class