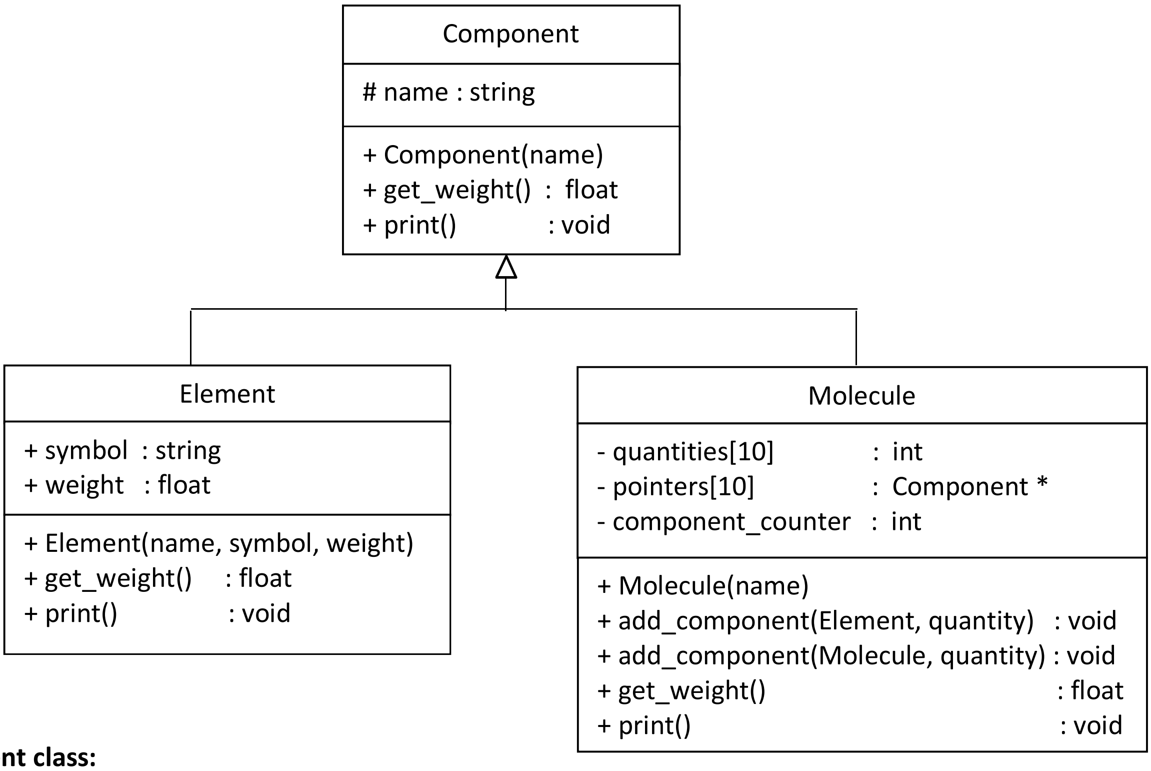
BLG252E – Object Oriented Programming
SAMPLE MIDTERM EXAM
Duration: 90 minutes. Books and notes closed.
(There are 3 questions , all are part of one problem.)

QUESTION 1) [65 points] Write the C++ classes whose inheritance hierarchy is shown in UML class diagram below.



Component class:
This class is an abstract base class which represents a generic entity.
The constructor takes the name for the component, which will be either an Element name or a Molecule name.
The get_weight() and print() functions should be pure virtual.

Element class:
The constructor takes the name of a chemical element, its symbol, and its atomic weight.
The get_weight() returns the atomic weight of the Element.
The print() function displays the member data on screen.

Molecule class:
A molecule is a composition of certain elements and/or other molecules.
Example: Water is a molecule (H_2O) that contains 2 Hydrogen atoms and 1 Oxygen atom. In water molecule, Hydrogen quantity is 2, Oxygen quantity is 1. So the total weight of water molecule is = 2*1.0 + 1*16.0 = 18.0

Another example: Ammonium dicarbonate $(NaH_4)_2C_2O_3$ is a compund molecule. It contains 2 quantity of ammonium molecules (NaH_4) , and 1 quantity of carbonate molecule (C_2O_3).

Data members of Molecule class:

quantities[10]	This integer array contains the quantity of each component. (Maximum array size is 10.)
pointers[10]	This is an array (**polymorphic**) of pointers to component objects. (Maximum array size is 10.) Some pointers may be to Element objects, some pointers may be to other Molecule objects.
component_counter	This counter will be incremented by 1, whenever a new component is added to the molecule.

Member functions of Molecule class:

Constructor	Takes the name of the molecule. Also initializes the component_counter to zero.
add_component(Element, quantity)	This function takes an Element object and its corresponding quantity, then updates the quantities array and the pointers array (a new Element object should be dynamically allocated).
add_component(Molecule, quantity)	This function takes another Molecule object and its corresponding quantity, then updates the quantities array and the pointers array (a new Molecule object should be dynamically allocated). To prevent self-inclusion, the function should check if the parameter Molecule object is not the same as the molecule object itself.
get_weight()	This function calculates and returns the sum of the weights of all components (all Element and all sub Molecule objects) in the molecule.
print()	This function displays molecule name and all of the quantities array along with calling the print() function pointed by the pointers array.

QUESTION 2) [20 points] In global scope, do the followings:

1. Declare an array of Element objects, and initialize that array with the chemical elements data given below.

Hydrogen	H	1.0
Carbon	C	12.0
Oxygen	O	16.0
Sodium	Na	23.0

2. Write the independent function ***get_element()*** whose prototype is given below.

Element & get_element(string searched_symbol)	This function takes an element symbol, then searches the given symbol in the global array of Elements. If search is successful, function should return the Element object found. Otherwise function should display a warning message and program must be stopped. This function will be called from the main program.

QUESTION 3) [15 points] In main program, do the followings:

1. Declare a molecule object m1 named “Ammonium”
   - add 1 quantity of Na element to m1 (use get_element function)
   - add 4 quantity of H element to m1 (use get_element function)

2. Declare a molecule object m2 named “Carbonate”
   - add 2 quantity of C element to m2 (use get_element function)
   - add 3 quantity of O element to m2 (use get_element function)

3. Declare a molecule object m3 named “Ammonium dicarbonate”
   - add 2 quantity of m1 to m3
   - add 1 quantity of m2 to m3

4. Call print functions of m1, m2, m3.