BLG252E – SAMPLE MIDTERM EXAM ANSWERS

```
#include <iostream>
#include <cstring>
#include <stdlib.h>
using namespace std;

class Component{ // Abstract base class
  protected:
  string name;
  public:
  Component(string n) : name(n) {};
  virtual float get_weight() = 0; //Pure virtual
  virtual void print() = 0;       //Pure virtual
 };

 //-----------------------------------------------------------
class Element : public Component
{
  public:
  string symbol;
  float  weight;
  Element(string en, string sy, float we) : Component(en)
  {
    symbol = sy;
    weight = we;
  }

  float get_weight() { return weight;};

  void print()
  {
    cout << "Element: " << name << " ,
          Symbol: " << symbol << " , Weight: " << weight << endl;
  }
};

//-----------------------------------------------------------
class Molecule : public Component
{
  int component_counter;
  int quantity[20];
  Component * pc[20]; // Polymorphic array of pointers

  public:
  Molecule(string mn) : Component(mn)
  {
   component_counter=0;
  }

  void add_component(Element e, int q)
  {
    pc[component_counter] = new Element(e); // Compiler-provided copy constructor
    quantity[component_counter] = q;
    component_counter++;
  };
  void add_component(Molecule m, int q)
  {
      if (this == &m)
      {
            cout << "Error : A molecule can not include itself!\n";
            return;
      }

    pc[component_counter] = new Molecule(m); // Compiler-provided copy constructor
    quantity[component_counter] = q;
    component_counter++;
  };

  float get_weight()
  {
    float total_weight=0;
    int i;
      for (i=0;i<component_counter;i++)
      total_weight += quantity[i] * pc[i]->get_weight();

     return total_weight;
  };

  void print()
  {
      int i;
    cout << "Molecule: " << name << " , Counter: " << component_counter << endl;
    cout << "Total weight: " << get_weight() << "\n";
      for (i=0;i<component_counter;i++)
    {
      cout << "\nQuantity: " << quantity[i] << " \n";
      pc[i]->print();
      }
  }

}; // end of Molecule class

//-----------------------------------------------------------
// GLOBAL DECLARATIONS:
 Element elements[] = {
   Element("Hydrogen"  ,"H"  ,1.0),
   Element("Carbon"    ,"C"  ,12.0),
   Element("Oxygen"    ,"O"  ,16.0),
   Element("Sodium"    ,"Na" ,23.0)
 };

Element & get_element(string searched)
{
      int i,N;
    N = sizeof(elements)/sizeof(Element);
      for (i=0;i<N;i++)
      {
            if (searched == elements[i].symbol)
               return elements[i];
      }
      cout << "\n ** Element symbol " << searched << " not found!\n";
      exit(0); // program stop
}
```

```
//-----------------------------------------------------------
int main()
{
 Molecule m1("Ammonium");
 m1.add_component(get_element("Na"), 1);
 m1.add_component(get_element("H"), 4);

 Molecule m2("Carbonate");
 m2.add_component(get_element("C"), 2);
 m2.add_component(get_element("O"), 3);

 Molecule m3("Ammonium dicarbonate");
 m3.add_component(m1, 2);
 m3.add_component(m2, 1);

 m1.print();
 cout << "=============================\n";
 m2.print();
 cout << "=============================\n";
 m3.print();

 system("pause");
 return 0;
}
```