2013.2 Object-Oriented Programming and Design
Final Exam (Dec. 18th 7pm-8:20pm)

supervisor signature	

StudentID# : ( ) , Name : ( )

* You may answer in either Korean or English. As an exception, you can use only English words in problem 1.

1. (32points) Complete following sentences by filling out blanks (a)~(f) with the most appropriate English words.
You can use only English words in this problem 1. Otherwise, you will get some penalty.

(1) ([a]) means determining the exact implementation of a request based on both the request (operation) name and the receiving object at ([b]) time. (same as [a]) allows us to ([c]) new classes to existing systems without ([d]) the existing code.

(2) In a package diagram, packages are usually organized to maximize ([e]) within each package and to minimize ([f]) among packages.

(3) ([g]) provide common interface to step through the elements of any arbitrary type STL containers.

(4) Properties of abstract class :
- An abstract class contains at least one ([h]).
- No ([i]) of an abstract class can be created
- An abstract class can only be used as ([j]).

(5) UML diagrams can be categorized into three types of diagrams : ([k]) diagram, ([l]) diagram, and ([m]) diagram.

(6) Just like arrays, STL vectors use ([n]) storage locations for their elements, which means that their elements can also be accessed using offsets on regular pointers to its elements, and just as efficiently as in arrays. But unlike arrays, their size can change ([o]), with their storage being handled automatically by the container.

(7) In operator overloading, if operator functions are non-member functions, the operator functions must be ([p]) to access private or protected data (member variables).

2. (12points)
(1) What is a container class? Explain with sufficient details.
( )

(2) What is the difference between “virtual function” and “pure virtual function”? Explain with sufficient details.
( )

(3) What is the purpose of using an abstract class? Explain with sufficient details.
( )

3. (18points) What is the output of the following C++ program to the screen?

```
#include <iostream>
using namespace std;

class B {
public:
  B() { z=-2; }
  B(int z_val) : z(z_val) {}
  virtual int get_val() { return (z-1); };
  int gv2() { return (z-2); }
protected :
  int z;
};

class D1 : public B {
public:
  D1() { x=6; }
  D1(int x_val): x(x_val) {}
  int get_val() { return x; };
protected:
  int x;
};

class D2 : public B {
public:
  D2() { y=3; }
  D2(int y_val): y(y_val) , B(y_val) {}
  virtual int gv2() { return y*y; };
  int get_val() { return y; }
protected:
  int y;
};
```

```
int main()
{
  B Zero(0);   D1 One(1);   D2 Two(2);
  B* B_ptrArray[2];
  D2* D2_ptrArray[2];

  B_ptrArray[0] = &One;
  B_ptrArray[1] = &Two;
  D2_ptrArray[0] = new D2 ;
  D2_ptrArray[1] = &Two;

  cout << "1 : " << One.get_val() << endl;
  cout << "2 : " << B_ptrArray[0]->get_val() << endl;
  cout << "3 : " << B_ptrArray[1]->gv2() << endl;
  cout << "4 : " << D2_ptrArray[0]->gv2() << endl;
  cout << "5 : " << D2_ptrArray[1]->get_val() << endl;
  cout << "6 : " << Two.gv2() << endl;
  delete D2_ptrArray[0];
  return 0;
}
```

```
Output : (PUT YOUR ANSWER HERE)
```

4. (18points) Consider following C++ code and its execution input/output result. This program computes and displays an average value of input float-type values. Fill out blanks (a), (b), (c) and (d) with appropriate C++ codes.

```
#include <vector>
#include <iostream>
using namespace std;

// sum adds the values of the vector it is passed.
float sum( (a) )
{
  (b)
}
```

```
int main() {
  (c)  // Declare a vector 'a'.
       // type of vector element is float
   float temp;
   while (cin >> temp) {
      a.push_back(temp);
   }
   cout << "Average = "
        << (d) << endl;

   return 0;
}
```

```
Input :
1.0 1.5 2.0

Output :
Average = 1.5
```

5. (20points) C++ code below shows generic stack implementation using template. Fill out blanks (a)~(d) with appropriate codes.

```
template<typename T>
class Stack {
   int size;
   int top;
   T *stackPtr;
public:
   Stack(int n) { size=n; top=0; stackPtr=new T[size]; }
   ~Stack() { delete[] stackPtr; }
   bool push( (a) ) {
   // return true if push is successful
   // return false if the stack is full
      (b)
   }
   bool pop( (c) ) {
   // return true if pop is successful
   // return false if the stack is empty
      (d)
   }
   bool isEmpty() { if (top<=0) return true; else return false; }
   bool isFull() { if (top>=size) return true; else return false; }
};
```

```
#include <iostream>
int main()
{
   int x, y;
   float xf, yf;
   Stack<int> s1(5);
   Stack<float> s2(5);
   s1.push(5); s1.push(8);
   s1.pop(x);  s1.pop(y);
   s2.push(5.3); s2.push(8.1);
   s2.pop(xf);  s2.pop(yf);
   std::cout <<x<< " " <<y<< std::endl;
   std::cout <<xf<< " " <<yf<< std::endl;

    return 0;
}
```

```
Output :
8 5
8.1 5.3
```