2017.2 Object-Oriented Programming
Final Exam (Dec. 13th 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. (20 points) Complete following sentences about C++ by filling out blanks (a)~(j) with the most appropriate English words.

(1) (a.   ) function of a class has the right to access non-public members of the class.

(2) Every derived class of (b.   ) class must implement the inherited (c.   ) if objects are to be created from it.

(3) (d.   ) occurs when two or more methods (i.e. member functions) in one class have the same method name but different parameters.

In contrast, (e.   ) means having two methods with the same method name and the same parameters (i.e. method signature). One of the methods is in the parent class and the other is in the child class.

(4) UML stands for (f. U___________) (g. M____________) (Language).
(same as (f)): UML has become a world (h.   ),
(same as (g)): UML describes a software system at a high level of (i.   ),
Language: UML expresses an idea.

(5) (j.   ) member variables of a class are shared by all instances of the class.

2. (10 points) Explain the main difference between 'private' and 'protected' for access control in a C++ class.
(i) private : (   )

(ii) protected : (   )

3. (10 points) What are the main similarities and differences between an array and a vector.
(1) main similarity 1: [In terms of data storage location] (   )
main similarity 2: [In terms of memory access] (   )

(2) main differences: (   )

4. (10 points) Fill out empty blanks with appropriate words.
(1) What is a 'container class'? (   )

(2) What are examples of STL container classes?
List at least three examples. (a.   ) , (b.   ) , (c.   )

5. (12 points) Consider following C++ code and its execution output result. Insert appropriate code in (a) and (b).

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

class Student
{
public :
    Student(int id , char* name) {
        stu_id = id;
        stu_name = name;
    }
    ~Student() {}
    static bool comp_student(const Student& a,
                             const Student& b)
    {
     (a)

    }
    char* getName() { return stu_name; }
    int getID() { return stu_id; }
private :
    int stu_id;         // student id
    char* stu_name;     // student name
};
```

```
(b)  // insert your code for Display function implementation




int main() {
          vector<Student> stu_vec;
          stu_vec.push_back(Student(4,"Nancy"));
          stu_vec.push_back(Student(1,"Tom"));
          stu_vec.push_back(Student(3,"Mike"));
          stu_vec.push_back(Student(2,"Lisa"));
         // sort by student id
          sort(stu_vec.begin() , stu_vec.end(), Student::comp_student);
          Display(stu_vec.begin() , stu_vec.end());

          return 0;
}
```

Execution Output result:

```
1 : Tom
2 : Lisa
3 : Mike
4 : Nancy
```

6. (10 points) What is the output of the following C++ program to the screen?

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

class B {
public:
  B() { z=-7; }
  B(int z_val) : z(z_val) {}
  virtual int get_val() { return (z-2); };
  int gv2() { return (z-3); }
protected :
  int z;
};

class D1 : public B {
public:
  D1() { x=9; }
  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) {}
  ~D2() { cout << “destructor called! (y=” << y << “)” <<endl;
}
  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)

7. (12 points) C++ code below shows generic stack implementation using template. Fill out blanks (a)~(c) with appropriate codes. In (c), insert your code for **push** member function.

```
template<typename T>
class Stack {
   int size;
   int top;
   T *stackPtr;
public:
   Stack(int n) { size=n; top=0; (a)                                       }
   ~Stack() { delete[] stackPtr; }
   bool push( (b)                   ); // return true if push is successful
                                       // return false if the stack is full
   bool isFull() { if (top>=size) return true; else return false; }
   ...   // we assume pop() and isEmpty() member functions were well-implemented.
};
```

(c) **// Insert your code for push member function**

```
#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
```

8. (10 points) Explain the main advantage and disadvantage of ‘call-by-value’ method by filling out following empty parentheses.

(i) advantage: (   )

(ii) Explain why such advantage can be obtained: (   )

(iii) disadvantage: (   )

(iv) Explain why such disadvantage can be obtained: (   )

9. (6 points) What are the possible purposes of using ‘&’ (reference type) for a function parameter (e.g. **void f(T& arg2)**)? There are two important purposes (that are different from the purposes of using call-by-value method). Explain.

(purpose 1:   )

(purpose 2:   )