# 2010.2 Object-Oriented Programming and Design
## Final Exam (Dec. 21th 5pm-6:30pm)

| | |
|---|---|
| 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. (18points) Complete following sentences by filling out the blanks with the most appropriate English words.
You can use only English words in this problem 1. Otherwise, you will get some penalty.

(1) STL vector is able to (a.                                        ) itself automatically when inserting a new vector element.

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

(3) Three main components of STL are (f.                              ) , (g.                              ), and (h.                              ).

(4) (i.                                        ) is a function or method whose behavior can be overridden within an inheriting class by a function with the same signature.

2. (10points) Describe why UML is useful? List four main reasons.

(1) (                                                                                )

(2) (                                                                                )

(3) (                                                                                )

(4) (                                                                                )

3. (12points)

(1) What is "abstract class"? Explain in detail.

(                                                                                )

(2) What is the purpose of using abstract class? Explain in detail.

(                                                                                )

(3) What is "container class"? Explain in detail.

(                                                                                )

4. (10points)

(1) What is the main similarity and difference of "array" and "STL vector"? Explain

(a) similarity : (                                                                                )

(b) difference : (                                                                                )

(2) What is the main similarity and difference of "overloading" and "overriding" in inheritance? Explain.

(a) similarity : (                                                                                )

(b) difference : (                                                                                )

5. (16points) In our project#3 sample code, **Display** function was given as follows.

(1) Explain what "**timeDelta**" value means in line 1?

( )

(2) Explain why "**timeDelta**" value is necessary in `g_sphere[i].ballUpdate(timeDelta)` function call of line 10?
Your answer should include the explanation on how `ballUpdate` function uses "`timeDelta`" value for its purpose.

( )

```
1: bool Display(float timeDelta)
2: {
3:   int i=0, j = 0;
4:   if( Device)  {
5:     Device->Clear(0, 0, D3DCLEAR_TARGET | D3DCLEAR_ZBUFFER,
6:          0x00afafaf, 1.0f, 0);
7:     Device->BeginScene();
8:
9:     for( i = 0; i < 4; i++) {
10:      g_sphere[i].ballUpdate(timeDelta);
11:      for(j=0;j<4;j++){g_legowall[i].hitBy(g_sphere[j]);}
12:    }
13:    for(i = 0 ;i < 4; i++){
14:      for(j = 0 ; j < 4; j++) {
15:        if(i >= j) continue;
16:        g_sphere[i].hitBy(g_sphere[j]);
17:      }
18:    }
19:    g_legoPlane.draw(Device, g_mWorld);
20:    for (i=0;i<4;i++) {
21:      g_legowall[i].draw(Device, g_mWorld);
22:      g_sphere[i].draw(Device, g_mWorld);
23:    }
24:    g_target_blueball.draw(Device, g_mWorld);
25:    g_light.draw(Device);
26:
27:    Device->EndScene();
28:    Device->Present(0, 0, 0, 0);
29:    Device->SetTexture( 0, NULL );
30:  }
31:  return true;
32:}
```

6. (20points) 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); }
private :
  int z;
};

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

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

```
int main()
{
  B Zero(0);
  D1 Two(1);
  B* B_ptrArray[3];
  B_ptrArray[0] = &Zero;
  B_ptrArray[1] = &Two;
  B_ptrArray[2] = new D2 ;

  cout << "0 : " << B_ptrArray[0]->get_val() << endl;
  cout << "1 : " << Two.get_val() << endl;
  cout << "2 : " << Two.gv2() << endl;
  cout << "3 : " << B_ptrArray[1]->get_val() << endl;
  cout << "4 : " << B_ptrArray[2]->get_val() << endl;
  cout << "5 : " << B_ptrArray[2]->gv2() << endl;

  delete B_ptrArray[2];
  return 0;
}
```

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

7. (14points) Write a C++ function "Swap" that takes two parameters x and y, and swaps the values of the two parameters (meaning it assigns the value of x to y and the value of y to x). Note that the types of **x** and **y** are the same but the type is a generic type. Therefore, you must use template to write the "Swap" function that can accept any type of parameters as shown in the following sample code.

```
#include <iostream>
int main()
{
   int a=3, b=4;
   float c=3.5 , d=2.3;
   Swap(a,b);
   Swap(c,d);
   std::cout << a << "," << b << "," << c << "," << d << "\n";
   return 0;
}
```

```
output :
4,3,2.3,3.5
```

**(Write your Swap function here using template.)**