2011.2 Object-Oriented Programming and Design
Midterm Exam (Oct. 24th 7pm-8:20pm)

supervisor signature	

StudentID# : ( ) , Name : ( )

* You may answer in either Korean or English.

1. (12 points) Complete following sentences by filling out the blanks (a)~(d) with the most appropriate words.

(1) Software crisis is the term used to describe the impact of rapid increases in computer power and the (a.) of the problems. Software technology could not follow the (a.).

(2) In object-oriented programming, a (b.) is a function whose behavior can be overridden within an inheriting class by a function with the same signature.

(3) Fred Brooks stated that over 90% of the costs of a typical system arise in the (c.) phase.

(4) Dynamic binding means that binding occurs at (d.).

2. (12 points) (1) What is portability(이식성)? Explain in detail.
( )

(2) What is dependability? Explain in detail.
( )

3. (14 points) (1) What is data encapsulation? Explain in detail.
( )

(2) What are two important benefits of data encapsulation? You also need to explain why you think so.
(1:) (Why? :)

(2:) (Why? :)

4. (12 points) (1) What is the meaning of "**protected**" visibility specifier in class definition. Explain in detail.
( )

(2) It is recommended to use "**protected**" only where it is really necessary. Why? Explain in detail.
( )

5. (15points) (1) What is inline function? explain in detail.
( )

(2) <u>In what situation</u>, can we get benefits of using inline function? List at least two situations.
(1:)

(2:)

(3) What benefit can we obtain by using inline function? Why?
(What benefit?)

(Why?)

6. (35points) The class for "complex number" (복소수) can be often defined using operator overloading. Write your implementation of member functions (1) operator+ and (2) operator<<.

In (1) and (2), you may use standard library in your code. Your code should be grammatically and logically correct.

```
class Complex
{ private:
        double re;  // real number part
        double im;  // imaginary number part
 public :
        Complex() { re=0; im=0; }
        Complex(double a, double b) { re=a; im=b; }
        Complex operator+(const Complex& x);
        friend ostream& operator<<(ostream &o, const Complex &x);
        // prints x in "a+bi" format. for example, if re=3.2 and im=2.3, then this fucntion should print 3.2+2.3i
        // if re=3.2 and im=-2.3, then this function should print 3.2-2.3i
        // if re=3.2 and im=0, then this function should print 3.2

        // other member functions should be here but they are not shown.
};
```

---

(1) Write C++ code for the **operator+** function

```
Complex Complex::operator+(const Complex& x)
{

}
```

(2) Write C++ code for the **operator<<** function.

```
ostream& operator<<(ostream &o , const Complex& x)
{

}
```

(3) Explain why **operator<<** function should be declared as **friend** of class complex in the above code.

( )

(4) Do we need to write a copy constructor for class **Complex**? If so, explain why. If not, explain why not.

( )

(5) In the member function **operator+(const Complex& x),**

what are possible benefits of using the parameter type **"const Complex&"**? List at least two benefits.

(1:)

(2:)