2018/05/22

Program Design
Second Mid-Test – 2018

I. Closing: finishing the sentence (30 %)

a) Class members are accessed via the dot (.) operator in conjunction with the name of an object (or reference to an object) of the class or via the Arrow (→) operator in conjunction with a pointer to an object of the class.

b) Class members specified as private are accessible only to member functions of the class and friends of the class.

c) ______ can be used to assign an object of a class to another object of the same class without overloaded.

d) ______ must be used to initialize **constant** members of a class.

e) A nonmember or global function must be declared as a(n) friend of a class to have access to that class's private data members.

f) A constant object must be initialized when it is created; it cannot be modified after it's created.

g) A(n) static data member represents class-wide information and operations.

h) An object's non-static member functions have access to a "self pointer" to the object called the this pointer.

i) Keyword const specifies that an object or variable is not modifiable.

j) If a member initializer is not provided for a member object of a class, the object's default ~~[illegible]~~ constructor is called.

k) A member function should be static if it does not access non-static class members.

l) Member objects are constructed ______ their enclosing class object.

m) Suppose **a** and **b** are integer variables and we form the sum a + b. Now suppose **c** and **d** are floating-point variables and we form the sum c + d. The two + operators here are clearly being used for different purposes. This is an example of ______.

n) Keyword ______ introduces an overloaded-operator function definition.

o) To use operators on class objects, they must be overloaded, with the exception of operators ______ , ______ and ______.

p) The ______ ,______ and arity of an operator cannot be changed by overloading the operator.

q) The operators that cannot be overloaded are ______,______, ______ and ______.

r) The ______ operator reclaims memory previously allocated by new.

s) The new operator dynamically allocates memory for an object of a specified type and returns a ______ of that type.

t) ______ is a form of software reuse in which new classes absorb the data and behaviors of existing classes and embellish these classes with new capabilities.

a) A base class's ______ members can be accessed in the base-class definition, in derived class definitions, and in friends of the base class and its derived classes.

u) In a(n) ______ relationship, an object of a derived class also can be treated as an object of its base class.

v) A base class's ______ members are accessible within that base class and anywhere that the program has a handle to an object of that class or one of its derived classes.

II. Complex numbers have the form: (40%, 5% each)

realPart + imaginaryPart * *i*

where ***i*** has the value $\sqrt{-1}$

b) Please create a class ***Complex***; use double type variables to represent the private data **realPart** and **imaginaryPart**.

c) Define a constructor that accept two arguments, e.g. 3.2, 7.5. to initialize the data members by using **member-initializer syntax**. Make this constructor a default constructor too by assigning the two data members both to values 1.0. The constructor also prints out a message like:

Complex number (3.2, 7.5) is constructed.

d) Define a destructor that prints a message like:

Complex number (3.2, 7.5) is destroyed.

e) Define a copy constructor that creates a complex number object and initializes by using another complex number object.

f) Overload the + operator to adds another complex number to **this** complex number object.

g) Overload both the << and >> operators (with proper friendship declarations) to output an Complex object directly and input two double values for a Complex object.

h) Overload the == and the != operators to allow comparisons of complex numbers. (please use definition of == to define !=)

i) Overload the ++ and the -- operators for **pre-** and **post-** operations that adds 1 to and minus 1 from both the **realPart** and the **imaginaryPart** of a Complex object.

III. Using class **Complex** in II, a **ComplexVector** class is defined as : (30%)

```
#include "Complex.h"
class ComplexVector
{
public:
  ComplexVector( int = 0 ); // default constructor
  ComplexVector( const ComplexVector & );
  ~ComplexVector(); // destructor

  const ComplexVector &operator=( ComplexVector & ) const;
private:
  int size;          // size of the vector
  Complex *ptr    // pointer to first element of vector
}; // end class ComplexVector
```

a) Implement the constructor which creates a vector of size indicated by its parameter and input the same number of complex numbers from input stream.

b) Implement the destructor.

c) Overload the assignment operator '=', which assign a ComplexVector object to another.

媽媽說：你看你房間跟豬窩一樣亂，還不趕快打掃……
兒子回答：妳有看過豬會打掃的嗎？不都是養豬的在打掃……
媽媽－﹏－'''