BLG252E – Object Oriented Programming
MIDTERM EXAM-2 (28.11.2013)
Duration: 90 minutes , Books and notes closed.

QUESTION 1) [15 points] What is the screen output when you execute the program below?

```
#define max(s1, s2)  s1>s2 ? s1 : s2

class A {
public:
  int x,y;
  A(int n=10, int m=20) {x=n; y=m;}
  int f1(){return max(x,y);}
  int f2(int n){x = max(x,n); return x;}
  int f3(int n){y = max(y,n); return y;}
};

class B : public A {
public:
  int z;
  int f1(){return x+y;}
  int f4(){z = x*x; return z;}
};
```

```
int main() {
  A a(40, 50);
  cout << a.f1()    << endl;
  cout << a.f2(60)  << endl;
  cout << a.f3(30)  << endl;
  cout << "-----\n";
  B b;
  cout << b.f1()    << endl;
  cout << b.f2(15)  << endl;
  cout << b.f3(5)   << endl;
  cout << b.f4()    << endl;
  cout << b.f1()    << endl;
  cout << b.A::f1() << endl;
  cout << "-----\n";
  a = b;
  cout << a.f1()    << endl;
  return 0;
}
```

QUESTION 2) [40 points] Consider the following UML class diagram for a bank account.

Account
- name : string - value : float
+ Account(cust_name, init_value) + deposit(amount) : void + withdraw(amount) : bool + withdraw(amount, feeRate) : bool + transfer(target, amount) : bool + print() : void

Function	Explanation
Constructor	The default is 0 for the init_value.
Deposit	Adds the given amount to value.
Withdraw / 1	Subtracts the given amount from value and returns true, if the given amount is less than or equal to value. Otherwise it returns false.
Withdraw / 2	Does the same operation, but an exra fee (to be applied to the given amount) is also subtracted from the value.
Transfer	The target is a reference to another Account object in which the given amount will be deposited. The given amount is withdrawn from the source account. A fixed fee of %1 is always deducted from the source object. Also, check whether target and source are the same object,and print an error message if so.
Print	Displays the name and value on screen.

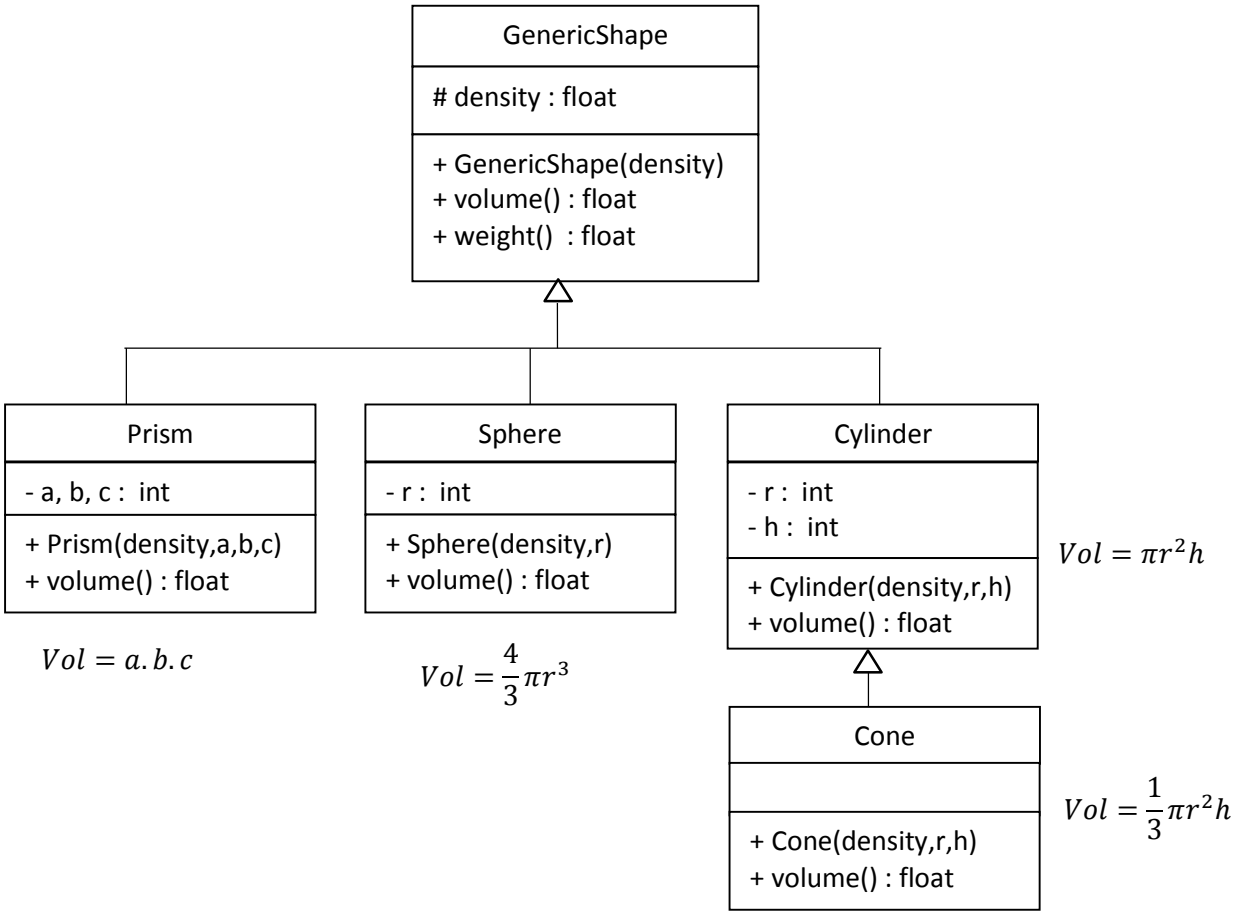
a) [30 points] Write C++ implementation codes for the Account class.

b) [10 points] Write the main() program with the testing operations below.

1. Declare M1 object with “AAA” as customer name and 1000 as initial value;
2. Withdraw 500 from M1 with %2 fee rate.
3. Deposit 100 to M1.
4. Declare M2 object with “BBB” as customer name.
5. Transfer 300 from M1 to M2.
6. Withdraw 200 from M2 without any fee.
7. Deposit 400 to M2.
8. Transfer 100 from M2 to M1.
9. Transfer 70 from M1 to M1.
10. Print M1 and M2.

QUESTION 3) [45 points] The following UML class diagram shows hierarchies and derivations of geometric shapes.

- GenericShape class is an abstract base class, and its volume() function should be <u>pure virtual</u>.
- The weight() function should simply call the volume() function, then multiply the volume with the density and return the result.



a) [30 points] Write C++ implementation codes for all classes.

b) [15 points] In your main() program do the followings:

- Declare an array of GenericShape pointers with 10 elements.
- For each array element, dynamically allocate an object with the constructor parameters data given below. (The pointer array will store memory addresses of different types of objects.)
- Looping through the pointers array, calculate and display the weight of each object.

	Array of GenericShape pointers	Object type	Constructor input parameters					
			density	a	b	c	r	h
0	→	Prism	2	6	8	5		
1	→	Prism	3	5	4	9		
2	→	Sphere	1				3	
3	→	Cylinder	1				4	8
4	→	Cylinder	4				2	9
5	→	Cone	3				5	7
6	→	Prism	1	7	2	1		
7	→	Sphere	2				9	
8	→	Cylinder	1				4	6
9	→	Cone	1				5	3