

2019.2 Object-Oriented Programming
Final Exam (Dec. 13th 7pm-8:20pm)

supervisor signature	
-------------------------	--

StudentID# : () , Name : ()

* You may answer only in English.

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

	call-by-value	call-by-reference
advantage	<ul style="list-style-type: none"> safety: call-by-value is often a safe method for argument passing because (a). 	<ul style="list-style-type: none"> The function can (b.) the value of the argument variable. Therefore, it is often quite useful. There is no (c.) of the argument made. Therefore, it is fast especially when passing large objects.

2. (15 points) Circle TRUE or FALSE for each of the following statements. If your answer is FALSE, make a correction to the wrong part of the statement. For correction, you should mark the wrong parts and modify the wrong parts into correct English words or expressions. Appropriateness of your correction will be considered. If your answer is TRUE, no correction is necessary.

(1) In C++, if a class contains one pure virtual function and one non-virtual function, it is correct to call the class an abstract class. (TRUE / FALSE)

If this is FALSE, make a correction to the statement : ()

(2) In C++, pointers to a base class can be assigned the address of a derived class object. (TRUE / FALSE)

If this is FALSE, make a correction to the statement : ()

(3) In C++ STL, a map is a sequential container. (TRUE / FALSE)

If this is FALSE, make a correction to the statement : ()

(4) The member function size() of C++ STL vector returns the size of allocated memory storage in bytes. (TRUE / FALSE)

If this is FALSE, make a correction to the statement : ()

(5) Compared to arrays, STL vectors consume more memory in exchange for the ability to manage storage and grow dynamically in an efficient way. (TRUE / FALSE)

If this is FALSE, make a correction to the statement : ()

(6) A derived class cannot have a method with the same name as a base class method. (TRUE / FALSE)

If this is FALSE, make a correction to the statement : ()

3. (15 points) Answer to each of following questions.

(1) In STL vector, elements are stored in contiguous (i.e. consecutive) storage locations.

The "being stored in contiguous storage locations" often has a big advantage. What is the most important advantage? Explain with sufficient detail.

()

(2) Arrays and STL vectors are similar in the sense that data elements are stored in contiguous storage locations. However, they have their own advantages. Answer to following questions.

(2-a) What can be the main advantage of using STL vectors compared to using arrays? Explain with sufficient detail.
()

(2-b) What can be the main advantage of using arrays compared to using STL vectors? Explain with sufficient detail.
()

4. (16 points) We can define a general max function that can take various types. Fill out empty boxes with appropriate C++ codes.

<pre>#include<iostream> using namespace std; // fill out empty boxes with appropriate C++ code using template. (a) (b) max((c)) { return (d); }</pre>	<pre>int main() { cout << max (3,5) << endl; cout << max (5.6, 4.3) << endl; cout << max ('k', 'a') << endl; return 0; } Output result: 5 5.6 k</pre>
--	--

5. (18 points) Consider following C++ code. What will be the execution output result? Insert your answer in the empty box below.

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

class A {
public:
    int f() { return 1; }
    virtual int g() { return 2; }
};

class B: public A {
public:
    int f() { return 3; }
    virtual int g() { return 4; }
};

class C: public A {
public:
    virtual int f() { return 7; }
    int g() { return 5; }
};
```

```
int main(){
    A *pa;
    A a;
    B b;
    C c;
    pa=&a; cout << pa->f() << endl; cout << pa->g() << endl;
    pa=&b; cout << pa->f() + pa->g() << endl;
    pa=&c; cout << pa->f() << endl; cout << pa->g() << endl;
    cout << b.f() << endl; cout << b.g() << endl;
    cout << c.f() << endl; cout << c.g() << endl;
    return 0;
}
```

// Execution Output result: insert your answer in this box

6. (24 points) Followings are two different **swap** function implementations that switch two X type input objects. One (left code) uses reference type parameters and the other (right code) uses pointer type parameters. Fill out the empty boxes with appropriate C++ code.

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

class X {
    int z;
public:
    X() { z=0; }
    X(int p) { z=p; }
    int getval() { return z; }
};

void swap( (a) ) // reference type parameters
{
    (b)
}

int main()
{
    X a(1), b(5);

    swap( (c) ); // (*)

    cout << "a=" << a.getval() << endl;
    cout << "b=" << b.getval() << endl;
    return 0;
}
```

Output result:
a=5
b=1

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

class X {
    int z;
public:
    X() { z=0; }
    X(int p) { z=p; }
    int getval() { return z; }
};

void swap( (d) ) // pointer type parameters
{
    (e)
}

int main()
{
    X a(1), b(5);

    swap( (f) ); // (#)

    cout << "a=" << a.getval() << endl;
    cout << "b=" << b.getval() << endl;
    return 0;
}
```

Output result:
a=5
b=1