Data Structures and Object-Oriented Programming

107 Spring

National Chiao Tung University

Time: 110 minutes

Instructions:

A. There are FIVE pages.
B. There are TEN questions. You must answer all of them.
C. Write your answers ONLY on the answer book.
D. DO NOT TURN OVER the cover page until you are told to do so.
E. You should assume that all the libraries are included properly for each question.
F. CHEATING IS A SERIOUS MATTER. YOU WILL RECEIVE A SCORE OF ZERO and SERIOUS PENALTY IF YOU CHEAT.

1.(a) [1+1+1+1%] After L1 is executed, what are the values of c.x, c.y, d.x, and d.y?

```
class Z {
    public:   Z(): x(-1), y(1) { }
                  int x, y;
};
void main( ) {
    Z c, d;
    d.x = ++(c.y) + (c.x)--;        // L1
    ... ...                         //extra lines of code here
}
```

1.(b) [1+1+1+1%] Given the following code fragment:

```
int *k( ) {
    return new int[4];              // L1
}
void h( ) {
    int *p = k(); *p = 8;
    *(&(p[2])+1) = 5;
    *(p+2) = 2;
    *(&p[1]) = 3;                   // L2
    ... ...                         //extra lines of code here
}
```

The function h() is called. What are the values of p[0], p[1], p[2], and p[3] after L2 is executed? Write down **unknown** if the value cannot be determined.

2. Given the following code fragment:

```
class Z {
public:   Z( ) { c = ' '; } // c is initialized to be a single white space character, i.e., 0x20.
              Z(char c) : c(c) { }
              char c; Z *next, *prev;
};
void main( ) {
    Z d('a'), e('b'), f;                                    // L1
    d.next = &e;   e.next = &f;   f.next = &d;
    d.prev = &f;   e.prev = &d;   f.prev = &e;
    f.prev->c = 'x';
    d.next->prev->c = 'p';                                  //L2
    f.next->next->next->prev->c = 'q';
    (*(d.prev->next->next->next)).c = 'y';                  // L3
    Z *k = &f;
    for (int i = 0; i < 123; ++i ) {
        k->c = '0'+i%10;
        k = k->prev;
    }                                                       // L4
    ... ...// other lines of code
}
```

2.a) [2%] After L1 is executed, what are the values of d.c and e.c?
2.b) [2%] After L2 is executed, what are the values of d.c and e.c?
2.c) [3%] After L3 is executed, what are the values of e.c and f.c?
2.d) [3%] After L4 (the for-loop) is executed, what are the values of d.c and f.c?

3. Answer the questions based on the following code fragment. The file content of sc.txt is as follows (in hexadecimal format; the memory address is increasing from left to right):

```
df 34 56 78 91 a5 b7 f8 15 e0 3d 10
```

The following code fragment is executed:

```
unsigned char x[3]; unsigned short y; int z;
fstream input;
input.open("sc.txt", ios::in|ios::binary );    input.seekg(8, ios::beg);
cout << input.tellg() << endl;                                    // L1
input.read(reinterpret_cast<char*>(&x[0]), sizeof(unsigned char));
cout << input.tellg() << endl;        input.seekg(-3, ios::cur);      // L2
input.read(reinterpret_cast<char*>(&x[1]), sizeof(unsigned char));
cout << input.tellg() << endl;        input.seekg(-6, ios::end);      // L3
input.read(reinterpret_cast<char*>(&y), sizeof(unsigned short));
cout << input.tellg() << endl;        input.seekg(2, ios::cur);       // L4
input.read(reinterpret_cast<char*>(&x[2]), sizeof(unsigned char));
cout << input.tellg() << endl;        input.seekg(-5, ios::end);      // L5
input.read(reinterpret_cast<char*>(&z), sizeof(z));
```

3.a) [1%x5] What is the output at each line, from L1 to L5?

3.b) [2%x5] What are the values of x[0], x[1], x[2], y, and z? Write the answers in the hexadecimal format. Assume that little endian is adopted for storing an unsigned short and an integer.

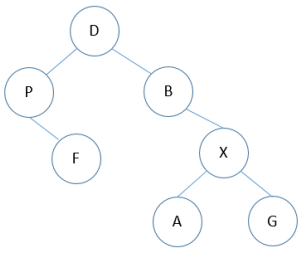
4.a) [1%x10] The following template class X implements an assignment operator *. A new object is created for storing the result. The default type is int. Fix the syntax and semantics errors. There are at least 10 mistakes (syntax and semantics). **If you modify a correct source code, there is a penalty, -1 point.**

```
template<datatype TD == int > class X {
    private:
    X( ):: 0(a) { }                          // initialize a to 0.
    X( int a) a(a) { }                       // initialize this->a to a
    X *operator*( X & const    b) {
        X &c;
        c->a = (*this).a*b->a;
        return (c);
    }
    <TD> a;
};
```

4.b) [1.5+1.5%] After the errors are fixed, write two different examples. Each example has at most five instructions to show: 1) initialize the data member of each object to a non-zero value; 2) use the operator to compute a result; and 3) store the result to an object.

5. [4%] Describe the major steps of merge sort. Use the following 8 elements of an array to show the steps: 5, 3, 2, 6, 1, 8, 7, 4. Sort the elements in ascending order. Draw the result of each step.

6. Given a binary tree as follows:



6.a) [2%] What is the pre-order traversal output?

6.b) [2%] What is the in-order traversal output?

6.c) [2%] What is the post-order traversal output?

6.d) [2%] What is the level-order traversal output?

6.e) [2%] What is the in-order traversal output for the subtree rooted at B?

7. Answer the questions based on the following code fragment.

```
class P {
public:    P( ) { c = 'P'; cout << c << endl;}
           P(char c) { (*this).c = c; }
           void printf() const { cout << "\t" << c << endl;}
           char c;
};
class Q: public P {
public:    Q( ) { c = 'q'; }
           Q (char c) { this->c = c; cout << c << endl;}
           virtual void printf() const { cout << "Q:" << "\t" << c << endl;}
           char c;
};
class R: public Q {
public: R( ):Q('q') { cout << "c:" << P::c << endl; }
           R(char name): Q( ) { this->c = c; cout << c << endl;}
           virtual void printf() const { cout << "R:" << "\t" << P::c << endl;}
};
void main( ) {
      R x;              // L1
      Q y('b');         // L2
      P z('c');         // L3
      Q *a = &x;
      a->printf( );     // L4
      a = &y;
      a->printf( );     // L5
   ……
}
```

7. [2%x5] Write down the output for each line (from L1 to L5) after the line is executed. If there is no output, write down **no output**. The output format must be correct, e.g., including new lines, tab, etc.

8. [10%] Draw a binary tree which satisfies the two conditions as follows:

The output for the level-order traversal: B F D C G E A

The output for the post-order traversal: F A G E C D B

9 . Given the following code fragment:

```
class A {
    static int cc;
     public:
     A( ) { cc++; cout << "cA" << endl;}          ~A( ) { cc--; cout << "dA" << endl;}
};
class B: public A {
     public:
     B( ) { cc += 2; cout << "cB" << endl;}       ~B( ) { cc -= 2; cout << "dB" << endl;}
};
class C : public B {
     public:
     C( ) { cc += 3; cout << "cC" << endl;}       ~C( ) { cc -=3; cout << "dC" << endl;}
};
int A::cc = 0;
void main( ) {
     A *a;              // L1
     B b;               // L2
     C *c;              // L3
     a = new B;         // L4
     c = new C;         // L5
     delete c;          // L6
     ……                 // extra lines of code here
}
```

9.a) [1%+2%+2%+2%+1%+1%] What are the output after L1, L2, L3, L4, L5, and L6 are executed? The output format must be correct, e.g., including new lines, tab, etc. If there is no output, write down **no output**.

9.b) [1%] After Line L6 is executed, what is the value of A::cc?

10.a) [3%] Write down in big-O notation for each of the following expressions:

i) $3n^2 + 1000000\ k\ n$; ii) $500 + 0.01n + 10^{-10}\ k\ 2^n$; and iii) $10000^k\ n + n \log n$.

Here, n is the input size and k is a positive constant.

10.b) [2%] Define a template function **min** that returns the minimum value of three arguments of the same type. Use at most four statements in the function body. **Deduce 1 point per extra statement.**

10.c) [3+2%] Write an example for exception handling. In the example, the exception argument is a string. Explain how the example works.

;END EXAM