清華大學 電機工程學系
107 學年度第二學期
EE-2310 計算機程式設計 (Introduction to Programming) 期末考試題
本試題 - 共計兩頁，九大題，總分 100 分

Closed-Book Examination (考試日期: Jan. 10, 2019)

1. (20%) Answer the following questions briefly.

(a) Is it true that a *constructor* function of a *class* should have a return type? (5%) ➔ No. Constructor should not have a return type.

(b) Consider an *union structure* defined below. If "double" has 8 bytes, then what is the size of an object of such an union structure? (To save space, we have compress several statements in a single line of code). (5%) ➔ 8 bytes

```
union PaySource {
    double hourly_pay;
    double monthly_salary;
}
```

(a) What is the maximum number of comparisons that a *binary search function* could possibly make when searching for a value in a 2000-element array? (5%) ➔ The answer is 11, since $\log_2(2000) < 11$

(b) Give a reason why *selection sort* could be more efficient than *bubble sort* for large arrays of big elements? (Note that a big element means an element with many data, for example, a *student record* containing a lot of information about a student). (5%) ➔ The bubble sort tends to have a lot more data movements!

2. (20%) Answer the following questions briefly.

(a) Are the members of a class *public* or *private* by default? (5%) ➔ private

(b) A class *Child* is a *derived class* of another class *Parent* by a statement shown below. If class *Parent* has a protected member, *p_data*, a public member function, *get_data*(), and a private data, *secret_data*. Then, what members of class *Parent* will be inherited by class *Child*? (5%) ➔ *p_data* and *get_data*()

class *B*: public *Parent* {…};

(c) Consider the following program segment. What will be displayed? (Note that the array name can be considered as a "pointer type of variable" pointing to the starting address of the entire array.) (5%) ➔ p[1]=A[2], Therefore, 3 will be displayed.

```
int A[5]={1, 2, 3, 4, 5};
int *p = A + 1;
cout << p[1]; // Hint: p[1] is equivalent to certain element in array A
```

(d) Show the content of C-string *numStr* after the execution of the following code segment. (5%) ➔ 101

char *numStr*[4];

itoa(5, *numStr*, 2); // converting an integer to a string of digit with a specified radix

3. (10%) Consider the processing of an array.

(a) Implement a sub-routine to support *main* function shown below to compute the sum of an integer array. (Note: you can skip the header files, namespace declaration, and function prototype. Declare your own variables when needed. (5%)

```
main(){
    int A[5]={1, 2, 3, 4, 5};
    cout << sum_of_array(A, 5); // display 15
}
// a function or sub-routine to be implemented here
```

```
Void sum_of_array(int *A, int size)
{
    int i, sum;
    for(int i=0; i<size; i++)
        sum = sum + A[i];
    return(sum);
}
```

(b) For array A[5] defined above, what does the following statement display? (5%) ➔ 53

cout << A[4] << “ “ << ++A[1] << endl;

4. (10%) Consider the usage of class **vector** provided by Standard Template Library, STL.

(a) What statement can you use to declare a **vector** of C++ strings? Please name it as *S*. (5%) ➔ vector<string> *S;*

(b) Write a few statements to insert two words into this array, “Good ” and “Day!”, and then print out S[0] and S[1] in sequence on the screen. (Hint: You can use member function “push_back()” for inserting an element into a **vector**, and operator “[]” to access one of its element.) (5%) ➔ S.push_back(“Good ”); S.push_back(“Day!”); cout << S[0] << S[1];

5. (10%) Answer the following questions related to **file IOs**.

(a) What should be put in as the second argument in the following "file stream creating function", if the file is to be created for output in the binary mode? (5%)

fstream *cio_yours*("yourfile", ios::binary | ios::out);

(b) Let A is an array with two elements of *double*. Use one statement to write this entire array to file "yourfile" in the binary mode.

***cio_yours*.write((char*)(A), sizeof(double)*2);**

6. (10%) There are two errors in Line 6 and Line 17. Fix them by showing their correct statements.

⇨ Line 6: void init(double w, double h, double l){

⇨ Line 17: b1.show_box();

```
1.    class Box
2.    {
3.        private:
4.            double w; double h; double l;
5.        public:
6.            void init(double w, h, l){
7.                    this->w = w; this->h=h; this->l=l;
8.            }
9.            void show_box(){
10.               cout << w << " " << h << " " << l << " ";
11.           }
12.   };
```

```
13.   main()
14.   {
15.           Box b1;
16.           b1.init(1, 2, 3);
17.           show_box(b1);
18.   }

// printing “1 2 3” on the screen
after execution
```

7. (10%) Consider the following bubble sort function that sorts an integer array into an *ascending order.*

(a) Fill in the missing program segment. Note that you can call the sub-routine *swap* directly. (5%)

(b) What is the time complexity in the Big-O notation? (5%) ➔ $O(n^2)$

```
void bubble_sort(int *A, int size)
{
        int i, next_to_fix;
        for(next_to_fix=(size-1); next_to_fix >0; next_to_fix--)
                // Program_Segment_to_be_Filled_in
                for(i=0; i< next_to_fix; i++){
                        if(A[i] > A[i+1]) swap(A[i], A[i+1]);
                }
}
```

```
void swap(int &a, int &b)
{
        int tmp = a;
        a = b;
        b = tmp;
}
```

8. (10%) Write a recursive function for computing the *combinatorial number* of selecting *k* elements from *n* unique elements, namely *Comb*(*k*, *n*). For example, Comb(2, 4) = 6. (Hint: Comb(k, n) = Comb(k, n-1) + Comb(k-1, n-1)).

```
int Comb(int k, int n)
{
      if(m<=0 || n<=0 || m>n){ exit(-1); }
        if(m==n    ||    n==1)        return(1);
        if(m==1)        return(n);
        return( Comb(k, n-1) + Comb(k-1, n-1) );
}
```