清華大學 電機工程學系
108 學年度第一學期
EE-2310 計算機程式設計 (Introduction to Programming) 期末考試題 (Final Exam)
本試題－共計兩頁，九大題，總分 100 分
Closed-Book Examination (考試日期: Jan. 9, 2020)

參考解答
Reference Answers

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

(a) Consider the following statement. If the result of variable *ans* is 1, then do it imply that string *s*1 and string *s*2 have the same content? (Simply answer YES or NO). (5%) ➔ NO

```
int   ans = strcmp(s1, s2);
```

(b) Complete the following library function call with blank A so that it converts a positive integer stored in variable *n*, into a string (named *binary_code*) that represents its binary representation. For example, if *n* is 64, then the *binary_code* will become "1000000". (Hint: what to be filled in is a library function name.) (5%)

```
char binary_code[100];
   A   (n, binary_code, 2) ; // fill in blanks A
```

➔ itoa

(c) What will be displayed after the execution of the following code segment? (5%)

```
int value = 123;
cout << left << setw(5) << setfill ('*');
cout << value;
```

➔ 123**

(d) 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
```

2. (10%) Answer the following questions briefly.

(a) Are the following two C++ statements legal? (Simply answer YES or NO). ➔ NO (&value is pointer of *float* not *int*)

```
float value;   int *pointer = &value;
```

(b) Show how to declare a dynamically allocated integer array for 100 elements. Use "*ptr_array*" as the pointer variable for the array created as shown below. (5%)

```
int *ptr_array = Blank to be filled in;
```

```
int *ptr_array = new int[100];
```

3. (10%) Fill in the two blanks (i.e., *BlankA* and *BlankB*) in the following member function for a class, *complex*, used to overload operator "<<", while supporting cascaded output.

```
BlankA   BlankB<<(ostream& os, complex &x) { … };
```

```
ostream&  operator<<(ostream& os, complex &x) { … };
```

4. (10%) Complete the following class definition in C++, called *complex*, for complex numbers, containing two private *double type of* data, *real* and *imaginary*, and a public **constructor function**. Fill in the three blanks, A, B, and C.

```
class complex{
    private   :
            double real;
            double imaginary;
    public   :
            complex(double x=0.0, double y=0.0){
            real = x;      imaginary = y;
      }
};
```

5. (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, "Happy" and "NewYear", 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("Happy"); S.push_back("NewYear"); cout << S[0] << S[1];

6. (10%) What will be displayed on the screen after the execution of the two following code segments, respectively.
   (a) Code Segment 1. (5%)

```
char s[15] = "Abracadabra";
char *found = strstr(s,"dab");
cout << found;
```

   ➔ dabra
   (b) Code Segment 2. (5%)

```
char world[1000]="Welcome! to the world of Programing in C++…";
char *token = strtok(world, " !");
cout << token;
```

   ➔ Welcome

7. (10%) Answer the following questions for sorting.
   (a) What is the time complexity of "selection sort" algorithm for *n* elements? Give your answers from {$O(\log_2 n)$. $O(n)$, $O(n*\log_2 n)$, $O(n^2)$}. (5%) ➔ $O(n^2)$
   (b) For an integer array A with 10 elements, how can we sort it by invoking a generic sorting function? Complete the funtion call, assuming that the function for comparing the order of two elements has been defined and named *mycompare():*

sort(A, A+10, mycompare); // function call to sort array A

8. (10%) Complete the following **recursive** function used to perform **binary search**. The arguments used are as follows: *A* is the integer array where the search is performed, *left* and *right* are the left and right indexes of the range of search, and *x* is the value of the target element under search.

```
int recursive_binary_search(int *A, int left, int right, int x)
{
   if(left>right) return(-1);
   int m = (left+right)/2;
   if(A[m]==x) return(m);
   if(x > A[m])
      recursive_binary_search(A, m+1, right, x);
   else
      recursive_binary_search(A, left, m-1, x);
}
```

9. (10%) Fix two errors in the following function that prints out all integers in an input file stream object.

```
void print_file(ifstream& in)
{
      int x;
      seekg(0, in::beg); // rewind to the very beginning of the file stream object
      while(in << x)       cout << x << endl;
}
```