2023.2 Object-Oriented Programming
Final Exam (Dec. 15, 7pm-8:20pm)

supervisor signature	

StudentID# : ( ) , Name : ( )

* You should write your answer in English. Otherwise, penalty may be applied.

1. (22 points) Complete following sentences about C++ by filling in blanks (a)~(k) with the most appropriate English words.

(1) (a.), also known as a function object, is an object that is able to be called as if it were a function.

(2) One of the main purpose of abstract class is to delegate (b.) responsibility to the derived class.

(3) An abstract class contains at least one (c.).

(4) We cannot create (d.) for an abstract class.

(5) (e.), which is defined in C++ STL algorithms library, applies an operation sequentially to the elements of one or two ranges and stores the result in another range.

(6) UML stands for (f. U_____) (g. M_____) (Language).
(same as (f)): UML has become a world (h.),
(same as (g)): UML describes a software system at a high level of (i.),
Language: UML expresses an idea.

(7) (j.) allows us to add new classes to existing systems without modifying the existing code.

(8) (k.) is a holder object that stores a collection of other objects with any data types.

2. (12 points) What are the relative advantage and disadvantage of array and STL vector compared to each other? Why? Explain.

(i) relative advantage of array compared to vector: (your answer should contain less than 8 English words.)

why? (your answer should contain less than 8 English words.)

(ii) relative disadvantage of array compared to vector: (your answer should contain less than 8 English words.)

why? (your answer should contain less than 8 English words.)

(iii) relative advantage of vector compared to array: (your answer should contain less than 8 English words.)

why? (your answer should contain less than 8 English words.)

(iv) relative disadvantage of vector compared to array: (your answer should contain less than 8 English words.)

why? (your answer should contain less than 8 English words.)

3. (15 points) Consider following C++ code.

```
==========================================================================
1:  int x=8;
2:  int* y;
3.  y=&x;
4:  int& z=x;
5:  int& w=z;
6:  w=*y*x;
7:  z++;
8:  int u = w;
9:  *y=x+5;
10: w = x + *y;
11: std::cout << "u=" << u << " x=" << x << " *y=" << *y << " z=" << z << " w=" << w << std::endl;  // print the values
==========================================================================
```

(1) What is the meaning of '*****' in line 2 ? ( )

(2) What is the meaning of '**&**' in line 3 ? ( )

(3) What is the meaning of '**&**' in line 4 ? ( )

(4) What is the meaning of the first '*****' and the second '*****' in '***y*x**' in line 6 ?
the meaning of the first '*****': ( ), the meaning of the second '*****': ( )

(5) What is the output result of executing line 11? ( )

4. (19 points) Consider following C++ program code and its execution input/output result. The function **sum_and_modify()** computes the sum of the values of the input vector **a** and modifies each element value of the input vector into **-1.5** . Insert appropriate code in empty boxes (a)~(d) such that this program generate the **output(##)** result from the given **input(#)**.

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

// compute the sum of the values of the input vector
// and modify each element value of the input vector into -1.5
double sum_and_modify( (a) ) {

  (b)

}
```

```
int main() {
    vector<double> a;   // Declare a vector.
    double temp;
    while (cin >> temp) { a.push_back(temp); }

    cout << "Average = " << (c)
         << endl;   // display the average of input values

    (d) it;
    for (it=a.begin(); it!=a.end(); it++) cout << *it << " ";
    cout << endl;
    return 0;
}
```

Input(#) :

```
1.1 1.2 1.3 1.4
<END-OF-FILE>
```

Output(##) :

```
Average = 1.25
-1.5 -1.5 -1.5 -1.5
```

5. (12 points) We can define a general **Max** function that can take various types using template. Fill in empty boxes (a)~(d) with appropriate C++ code.

```
#include<iostream>
using namespace std;
// fill in empty boxes with appropriate C++ code using template.
(a)
(b) Max( (c) )

{
    return (d) ;
}
```

```
int main()
{
    cout << Max (3,5) << endl;
    cout << Max (5.6, 4.3) << endl;
    cout << Max ('a', 'k') << endl;
    return 0;
}
```

Output result:

```
5
5.6
k
```

6. (20 points) What is the output of the following C++ program to the screen? Insert your answer into the empty box below.

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

class B {
public:
  B() { z=-9;  }
  B(int z_val) : z(z_val) {  }
  virtual int get_val() { return (z-2); }
  int gv2() { return (z-4); }
private :
  int z;
};

class D1 : public B {
public:
  D1() { x=3; }
  D1(int x_val): x(x_val) { }
  virtual int get_val() { return x; }
  int gv2() { return x+8; }
private:
  int x;
};

class D2 : public B {
public:
  D2() { y=6; }
  D2(int y_val): y(y_val) {}
  int get_val() { return y; }
  virtual int gv2() { return y*y; }
private:
  int y;
};
```

```
int main()
{
  B Zero(1);   D1 Two;  D2 Three;  B* d1ptr; D2*  d2ptr;
  B* B_ptrArr[3];
  B_ptrArr[0] = &Zero;  B_ptrArr[1] = &Two; B_ptrArr[2] = &Three;
  d1ptr = new D1(7);
  d2ptr = new D2 ;
  cout << "0: " << B_ptrArr[0]->get_val() << endl;
  cout << "1: " << B_ptrArr[0]->gv2() << endl;
  cout << "2: " << Two.get_val() << endl;
  cout << "3: " << Two.gv2() << endl;
  cout << "4: " << B_ptrArr[1]->get_val() << endl;
  cout << "5: " << B_ptrArr[1]->gv2() << endl;
  cout << "6: " << d2ptr->gv2() << endl;
  cout << "7: " << d2ptr->get_val() << endl;
  cout << "8: " << B_ptrArr[2]->get_val() << endl;
  cout << "9: " << d1ptr->get_val() << endl;
  delete d1ptr;  delete d2ptr;  return 0;
}
```

```
Output : (INSERT YOUR ANSWER HERE)
```