## Exercise 5: Template and Inheritance

**1. Purpose and requirements**

**Purpose:** Familiar with Template and Inheritance, including: (1) Template; (2) Class template; (3) Function template;

**Requirements:** simplicity, clarity, and user friendly

**2. Experiment contents：**

(1) Write a function template to reverse the elements using at least five ways, such as reverse the first n elements, reverse the elements of a given part, reverse the elements of two separate parts, or reverse the elements in your own way (function object). Then, give two non-template functions (function overloading). Finally, test them in your program. (make sure your program is user friendly and clear ☺)

Such as, template < class T> void reverse(T a[], int n);

void reverse(double \*a, int n);

(2) Write a class template stack<T>, to store the int, double, and string, test it! Don’t forget to make your program beautiful!

const int Max=20;

template < class T>

class stack

{

T s[Max]; //store data

int index; // the element index

public:

\* push(\*); //\* means you have to modify

\* pop(\*); //remove the last element, and save it in x.

\* getElement(\*);

\* stackEmpty(\*); //if empty return 1, or return 0

…. // add your own function or operator if you have good ideas, to make //your program like “math” ☺

};

(3) Write a class template to record something, such as fruit and its weight or price, students and their Ids, give proper interface to access these information. Imagine you are a super designer in a big company. Let me see your talent!

**3. Questions：**

1. Do you know when to define a function template?

2. How to inherit the template class?