Discussion 9
C++ Templates

Yanbing Dong
2016.5.3

Class

A user defined data type

```
class myClass
{
        int a, b
    public:
        myClass(){ . . .}
        myClass(int, int){ . . .}
        int add(){ . . .}
} ;

myClass c_int;
```

Then a class for float?

Class Template

A generic class for different types

container in STL, vector

```
template <typename T> or <class T>
class myClass
{
       T a, b
    public:
       MyClass();
       myClass(T,  T);
       T add();
} ;

template <typename T> T myClass<T>::add() {return a + b;}

myClass<int> c_int;     // template instantiation
myClass<float> c_float;
```

http://feihu.me/blog/2014/the-origin-and-usage-of-typename/

Something to consider

- put template declaration and definition in one header file

- Put them in one header file and one cpp file->more work->explicit instantiation
- in cpp file
-

```
template class myClass<int>;
```

Function Template

Generic function for different types   algorithms in STL, sort

```
template <typename T> T max(T, T);

template <typename T> T max(T a, T b)
{
      return a > b ? a : b;
}
```

Based on input to determine T ->implicit instantiation

```
int  x = max(6.5, 6.0);   // okay here
int x = max(6.5, 6);  // no matching function for call to ‘add(int, double)’
```

Something to consider

1. same as class template

Declaration and Definitin in one header file or

Explicit instantiation in cpp file

```
template double max<double>(double, double);
```

2. if T is not specified, then instantiate it explicitly

```
double tmp = 6.5;
template <typename T>
T get()
{
      return tmp;
}


cout << get() << endl;
```

Reference

http://www.is.pku.edu.cn/~qzy/cpp/vc-stl/templates.htm

https://docs.oracle.com/cd/E19205-01/820-1214/bkaew/index.html