



TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN TP.HCM
KHOA CÔNG NGHỆ THÔNG TIN
BỘ MÔN CÔNG NGHỆ PHẦN MỀM
MÔN: **LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG**

THỰC HÀNH TUẦN 11

TEMPLATE

TP.HCM, ngày 4 tháng 5 năm 2022

MỤC LỤC

1	Template	3
1.1	Method template	3
1.2	Class template	3
2	Bài tập.....	5

1 Template

1.1 Method template

Method template là phương thức đặc biệt không chỉ định rõ kiểu dữ liệu đối số truyền vào và kiểu dữ liệu trả về tại thời điểm định nghĩa mà việc này sẽ được thực hiện tại lời gọi phương thức, tham khảo ví dụ sau:

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

template<class T> T getMax(T a, T b);

int main(int argc, char* args[])
{
    int x = 20, y = 7;
    double a = 0.5, b=7.2;
    long n=200, m = 900;
    cout<< "So sanh int: " << (int)getMax(x,y)<< endl;
    cout<< "So sanh double: " << (double)getMax(a,b)<< endl;
    cout<< "So sanh long: " << (int)getMax(n,m) << endl;
    return 0;
}

template<class T> T getMax(T a, T b)
{
    if(a>= b)
        return a;
    return b;
}
```

Tại thời điểm định nghĩa hàm **getMax**, **T** là một kiểu dữ liệu không rõ ràng, khi gọi thực thi hàm getMax, chúng ta xác định kiểu dữ liệu và parameter truyền vào cho hàm.

1.2 Class template

Xét ví dụ sau:

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

template<class T>
class MyArray
{
private:
    T* array;
    int n;
public:
    MyArray();
    MyArray(int l);
}
```

```

~MyArray();
void setArray(T*,int);
void sort();
void print();
};

template<class T>
MyArray<T>::MyArray()
{
    array =NULL;
}
template<class T>
MyArray<T>::MyArray(int l)
{
    array =new T[l];
    n =l;
}
template<class T>
MyArray<T>::~~MyArray()
{
    if(array != NULL)
        delete[] array;
}
template<class T>
void MyArray<T>::setArray(T *a, int l)
{
    if(array != NULL)
        delete[] array;
    n = l;
    array= new T[n];
    for(int i=0;i<n;i++)
    {
        array[i] = a[i];
    }
}
template<class T>
void MyArray<T>::sort()
{
    T temp;
    for(int i=0;i<n-1;i++)
        for(int j=i+1;j<n;j++)
            if(array[i] < array[j])
            {
                temp =array[i];
                array[i] =array[j];
                array[j] =temp;
            }
}
template<class T>
void MyArray<T>::print()
{
    cout << endl;
    for(int i=0;i<n;i++)
        cout<< array[i] << " ";
    cout<< endl;
}

```

```
int main()
{
    int mangInt[5] = {4, -2, 5, 0, 6};
    double mangThuc[4]={6.0, -1.5, 0.5, -3};
    MyArray<int> ar1;
    MyArray<double> ar2;
    ar1.setArray(mangInt, 5);
    ar1.print();
    ar1.sort();
    ar1.print();

    ar2.setArray(mangThuc, 4);
    ar2.print();
    ar2.sort();
    ar2.print();
}
```

Lớp MyArray nhận vào mảng array mà kiểu dữ liệu không xác định trước.

2 Bài tập

Sinh viên khai báo đầy đủ lớp MyArray như trên và thực hiện các yêu cầu như sau:

Cho phép nhận vào bất kỳ kiểu dữ liệu nào mà người dùng mong muốn!?

- Sắp xếp các đối tượng mong muốn.
- Lấy ra một đối tượng trong mảng.
- Gán giá trị cho một đối tượng trong mảng.
- Sử dụng cơ chế Exception handling.

Sinh viên viết hàm main để test các chức năng của class MyArray với 2 kiểu dữ liệu là int và Phân số.