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

HƯỚNG DẪN THỰC HÀNH

LAB 05 – Class template

TP.HCM, ngày 25 tháng 10 năm 2016

MỤC LỤC

1 Mục tiêu

- Hiểu và sử dụng được Class Template

2 Đề bài

2.1 Khái niệm

Xét ví dụ xây dựng lớp mảng cho nhiều kiểu dữ liệu khác nhau.

```
// Mang so nguyen.
class IntegerArray
{
	private:
	int	*m_pElement;
	// Viet cai dat cho lop.
};

// Mang so thuc.
class FloatArray
{
	private:
	float	*m_pElement;
	// Viet cai dat cho lop.
};

// Mang ky tu.
class CharArray
{
	private:
	char	*m_pElement;
	// Viet cai dat cho lop.
};
```

Với mỗi kiểu dữ liệu khác nhau (int, float, char…) chúng ta phải xây dựng lớp mảng tương ứng cho kiểu dữ liệu đó. Điều này gây ra sự dư thừa không đáng có trong chương trình. Và hơn nữa các lớp trên vẫn không đủ dùng trong mọi trường hợp.

=> Dùng Class Template.

Class Template là lớp đối tượng tổng quát cho phép dùng nhiều kiểu dữ liệu khác nhau cho các thuộc tính và phương thức của lớp. Class Template được khai báo bắt đầu bằng từ khóa “template”.

```
template <class T> class
SampleClass

{

	// Viet cai dat.
```

```
};
```

2.1.1 Ví dụ

Để hiểu rõ hơn về Class Template, chúng ta xét ví dụ xây dựng lớp mảng cho nhiều kiểu dữ liệu khác nhau Array.

Bước 1: vào VC++, tạo project dạng Console Application.

Bước 2: thêm vào project file Array.h, viết khai báo và cài đặt cho lớp Array như sau:

```
template <class T> class Array
{
private:
	T	*m_pElement;
	int	m_iLength;

public:
	Array(int iLength)
	{
		if (iLength < 0)
		{
			cout << "Loi: chieu dai mang la so am.";
			return;
		}

		m_iLength = iLength;
		m_pElement = new T[m_iLength];
	}


	int GetLength()
	{
		return m_iLength;
	}


};
```

Bước 4: thêm vào project file main.cpp và viết đoạn chương trình sử dụng lớp Array vừa tao như sau:

```
#include "Array.h"

void main()
{
	Array<int>	a(3);

	a[0] = 0;	a[1] = 1;	a[2] = 2;

	for (int i = 0; i < a.GetLength(); i++)
		cout << a[i] << endl;
```

```
	Array<int> b(a);

	for (int i = 0; i < b.GetLength(); i++)
		cout << b[i] << endl;

	b[1] = 2;

	a = b;

	for (int i = 0; i < b.GetLength(); i++)
		cout << b[i] << endl;

}
```

Bước 5: Hãy viết tiếp mã nguồn cho `template <class T> class Array` để biên dịch và chạy thành công chương trình.

2.2 Bài tập tìm hiểu

Thử tách lớp template Array trong ví dụ thành 2 phần:

- Phần khai báo được đặt trong file Array.h
- Phần định nghĩa được đặt trong file Array.cpp

Vẫn tiếp tục sử dụng hàm main trong ví dụ. Hãy thử biên dịch, cho biết kết quả, giải thích và đề xuất phương pháp giải quyết.

3 Nộp bài

- Bài nộp phải được nén lại, đặt tên <**MSSV**>_Tuan05. Thư mục nộp được tổ chức như sau:

  <MSSV>_Tuan05: solution Visual Studio
- Xóa hết thư mục Debug, các tập tin *.ncb, *.sdf trước khi nén
- Cố gắng đảm bảo chương trình không bị lỗi biên dịch, các hàm, phương thức chưa cài đặt được vui lòng comment lại hoặc xuất ra nội dung **"Ham/Phuong thuc chua duoc cai dat"**

---Hết---