

BÀI 1

Lớp phân số (đặt tên là *PhanSo*) được viết để cho phép tính toán trên phân số. Đoạn chương trình sau đây được viết để chạy thử các phương thức và phép toán đã viết:

```
23  ...
24  int main() {
25      PhanSo a(1, 3), c;
26      int b = 8;
27      c = a + (PhanSo)b;
28      cout << "c = " << a << " + " << b << " = " << c << endl;
29      cout << "++c: ";
30      cout << "c = " << ++c << endl;
31      cout << "a = " << a << " + " << c << endl;
32      a += c;
33      cout << "a = " << a << endl;
34      b = a;
35      cout << "b = " << b << endl;
36      return 0;
37  }
```

- Giả sử các phép toán và phương thức của lớp *PhanSo* đã được viết đủ và đúng. Bạn hãy cho biết kết quả chạy của chương trình nói trên.
- Đọc kỹ từng dòng của mã nguồn (các dòng 24 – 37) để cho biết lớp *PhanSo* phải gồm tối thiểu những phương thức và phép toán nào. Viết tập tin *PhanSo.h* định nghĩa các phương thức cần thiết, như hướng dẫn trong bảng sau (chép lại mã nguồn và viết bổ sung đầy đủ):

```
// Tập tin PhanSo.h
#ifndef _PhanSo_h
#define _PhanSo_h
#include <iostream>
using namespace std;

class PhanSo {
private:
    int tu, mau;
public:
    PhanSo();
    // Các dòng mã cần viết thêm
    // ...
};
#endif
```

- Viết mã của 5 phương thức trong số các phương thức đã khai báo ở câu trên.

BÀI 2

- Trình bày về khái niệm đa hình (*polymorphism* – đa xạ) và cho biết ý nghĩa, tầm quan trọng của tính chất này trong lập trình hướng đối tượng. Viết đoạn mã nguồn để minh họa và diễn giải.
- Trình bày khái niệm của lớp trừu tượng (*abstract class*). Lớp trừu tượng được cài đặt trong C++ như thế nào? Hãy viết mã nguồn minh họa.

BÀI 3

<pre>#include <math.h> #include <stdio.h> class Function { public: virtual double Value(double x)=0; virtual double Derive(double x)=0; };</pre>	<pre>class Sin:public Function { public: double Value(double x) { return sin(x); } double Derive(double x) { return cos(x); } };</pre>
<pre>double DeriveProduct(Function* f, Function* g, double x) { if (f != NULL && g != NULL) { return f->Derive(x)*g->Value(x) + f->Value(x)*g->Derive(x); } return 0; }</pre>	
<pre>double DeriveSum(Function* f, Function* g, double x) { if (f != NULL && g != NULL) { return f->Derive(x) + g->Derive(x); } return 0; }</pre>	
<pre>void main() { double x = 3.14/6; // PI/6 Sin* s = new Sin(); double y = DeriveProduct(s, s, x); printf("sin(x) = %f; sin'(x) = %f\n", s->Value(x), s->Derive(x)); printf("y = %f\n", y); }</pre>	

Đoạn mã bên trên khai báo lớp *Function* nhằm mục đích cài đặt việc tính giá trị của một hàm số $f(x)$ (phương thức $Value(x)$) và giá trị của đạo hàm $f'(x)$ của nó (phương thức $Derive(x)$).

- Hãy giải thích việc khai báo các phương thức $Value(x)$ và $Derive(x)$ trong lớp *Function* (ý nghĩa của việc gán bằng 0 và từ khóa *virtual*).
- Hãy viết mã nguồn của lớp *Cos* tương ứng cho hàm $f(x) = \cos(x)$.
- Mã nguồn của hàm *DeriveSum()* nhằm mục đích cài đặt việc tính đạo hàm tổng dựa vào công thức tổng $(f + g)'(x) = f'(x) + g'(x)$ và mã nguồn hàm *DeriveProduct()* tính đạo hàm tích nhờ vào công thức tương ứng với tích $(f \cdot g)'(x) = f'(x) \cdot g(x) + f(x) \cdot g'(x)$. Bạn hãy viết các hàm cho phép tính:

- Đạo hàm của hàm thương dựa vào công thức: $\left(\frac{f}{g}\right)'(x) = \frac{f'(x)g(x) - f(x)g'(x)}{g^2(x)}$
- Đạo hàm của hàm hợp: $(f \circ g)'(x) = f'(g(x))g'(x)$.

- Mã nguồn của hàm *main()* bên trên có tạo một đối tượng của lớp *Sin* tức là hàm $\sin(x)$. Nếu gọi toán tử *new* nhiều lần thì sẽ có nhiều đối tượng của lớp *Sin* được tạo. Tuy nhiên trong thực tế ta chỉ cần một hàm $\sin(x)$ nên việc tạo ra nhiều đối tượng như thế là phung phí bộ nhớ. Bạn hãy đề xuất giải pháp và sửa mã nguồn của lớp *Sin* sao cho người lập trình chỉ có thể dùng một đối tượng duy nhất của lớp *Sin*.