TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHIỆP
KHOA ĐIỆN TỬ
BỘ MÔN TIN HỌC CÔNG NGHIỆP

TÀI LIỆU
LỜI GIẢI BÀI TẬP

TÊN HỌC PHẦN: **LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG C++**
MÃ HỌC PHẦN: **TEE319**

BIÊN SOẠN: **LÊ HẢI TRUNG**

THÁI NGUYÊN – 2015

LỜI NÓI ĐẦU

Lập trình hướng đối tượng là một phương pháp lập trình cho phép thao tác trực tiếp trên các đối tượng cụ thể, một chương trình viết theo hướng đối tượng được chia thành các lớp đối tượng từ đó cho phép các đối tượng này có thể kế thừa được những đặc tính của các đối tượng khác và hoạt động thông qua sự tương tác với các đối tượng khác nhờ cơ chế truyền thông báo. Từ đó giúp lập trình theo hướng đối tượng có được sự linh hoạt, tiện ích trong việc xây dựng và pháp triển các phần mềm.

Ngày nay lập trình hướng đối tượng được áp dụng rộng rãi trên thế giới, nó đã hầu như thay thế hoàn toàn các phương pháp lập trình truyền thống để mang lại sự hiệu quả cũng như tiện ích cho người sử dụng. Các ngôn ngữ lập trình hướng đối tượng được phát triển mạnh mẽ và được sử dụng thông dụng như C#, C++, Visual Basic, Java, Visual C... Vì vậy việc nghiên cứu phương pháp lập trình mới này là thực sự cần thiết đối với những người làm Tin học.

Cuốn bài tập này là tổng hợp lời giải của các đề bài cơ bản về lập trình hướng đối tượng trên C++. Đây là tài liệu tham khảo đối với các bạn sinh viên chuyên ngành CNTT, phục vụ tốt cho việc ôn thi và là cơ sở để luyện tập các kỹ năng lập trình hướng đối tượng trên những ngôn ngữ bậc cao hơn.

Trong quá trình biên soạn chắc chắn vẫn còn nhiều thiếu sót, hi vọng nhận được sự đóng góp ý kiến nhiệt tình của thầy, cô và các bạn.

MỤC LỤC

CHƯƠNG 1: CÁC KHÁI NIỆM CƠ BẢN

1.1 Tóm tắt lý thuyết 1

1.2 Các dạng bài tập 1

1.3 Các vấn đề về thảo luận, thực hành, thí nghiệm 1

1.4 Bài tập sinh viên tự làm 1

CHƯƠNG 2: LỚP (CLASS)

2.1 Tóm tắt lý thuyết 5

2.2 Các dạng bài tập 6

2.3 Các vấn đề về thảo luận, thực hành, thí nghiệm 7

2.4 Bài tập sinh viên tự làm 7

CHƯƠNG 3: TOÁN TỬ TẢI BỘI

3.1 Tóm tắt lý thuyết 28

3.2 Các dạng bài tập 28

3.3 Các vấn đề về thảo luận, thực hành, thí nghiệm 30

3.4 Bài tập sinh viên tự làm 30

CHƯƠNG 4: KẾ THỪA

4.1 Tóm tắt lý thuyết 43

4.2 Các dạng bài tập 44

4.3 Các vấn đề về thảo luận, thực hành, thí nghiệm 48

4.4 Bài tập sinh viên tự làm 48

CHƯƠNG 5: KHUÔN HÌNH

5.1 Tóm tắt lý thuyết 78

5.2 Các dạng bài tập 79

5.3 Các vấn đề về thảo luận, thực hành, thí nghiệm 80

5.4 Bài tập sinh viên tự làm 80

TÀI LIỆU THAM KHẢO

CHƯƠNG 1

CÁC KHÁI NIỆM CƠ BẢN CỦA LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG

1.1 Tóm tắt lý thuyết

- Tìm hiểu về cách tiếp cận hướng đối tượng, những ưu điểm, nhược điểm của lập trình truyền thống và các đặc điểm của lập trình hướng đối tượng.
- Các khái niệm cơ sở của phương pháp hướng đối tượng: Đối tượng, Lớp, Trừu tượng hóa dữ liệu và bao gói thông tin, kế thừa, tương ứng bội...
- Các bước cần thiết để thiết kế chương trình theo hướng đối tượng
- Các ưu điểm của lập trình hướng đối tượng
- Các ngôn ngữ hướng đối tượng

1.2 Các dạng bài tập

<u>Câu 1.1:</u> Tại sao phải lập trình hướng đối tượng?

Để làm được các bài tập của chương này sinh viên cần nắm chắc các khái niệm cơ bản, các đặc điểm, tính chất của lập trình hướng đối tượng,

1.3 Các vấn đề về thảo luận, thực hành, thí nghiệm

Không có

1.4 Bài tập sinh viên tự làm

<u>Câu 1.2:</u> Nêu đặc điểm của lập trình hướng đối tượng?

Lập trình hướng đối tượng có các đặc điểm quan trọng sau:

- Nhấn mạnh trên dữ liệu hơn là thủ tục
- Các chương trình được chia thành các đối tượng
- Dữ liệu được che giấu và không thể được truy xuất từ các hàm bên ngoài
- Các đối tượng có thể giao tiếp với nhau thông qua các hàm
- Dữ liệu hay các hàm mới có thể được thêm vào khi cần
- Theo tiếp cận từ dưới lên

Câu 1.3: Trong số các nhận định sau, nhận định nào đúng, nhận định nào sai:

- Đối tượng là một thực thể cụ thể, tồn tại thực tế trong các ứng dụng. → Đ
- Đối tượng là một thể hiện cụ thể của Lớp. → Đ
- Lớp là một khái niệm trừu tượng dùng để biểu diễn các Đối tượng. → Đ
- Lớp là một sự trừu tượng hoá của Đối tượng. → Đ
- Lớp và Đối tượng có bản chất giống nhau. → S
- Trừu tượng hoá đối tượng theo chức năng tạo ra các thuộc tính của lớp. → S
- Trừu tượng hoá đối tượng theo chức năng tạo ra các phương thức của lớp. → Đ
- Trừu tượng hoá đối tượng theo dữ liệu tạo ra các thuộc tính của lớp. → Đ
- Trừu tượng hoá đối tượng theo dữ liệu tạo ra các phương thức của lớp. → S
- Kế thừa cho phép hạn chế sự trùng lặp mã nguồn. → Đ
- Kế thừa cho phép tăng khả năng sử dụng lại mã nguồn. → Đ

Câu 1.4: Liệt kê tất cả các thuộc tính và hành động của đối tượng Xe ô tô. Đề xuất lớp Car (Ô tô).

- Thuộc tính: màu xe, nhãn hiệu, trọng lượng, giá thành...
- Hành động: khởi động xe, chạy xe, dừng xe, tắt máy.

```
class Car
{
      private:
            char colour[10];
            char brand[10];
            float weight, price;
       public:
            void start();
            void run();
```

```
            void stop();
            void shutdown();
    };
```

Câu 1.5: Liệt kê tất cả các thuộc tính và hành động của đối tượng Xe buýt. Đề xuất lớp Bus.

- Thuộc tính: màu xe, số chỗ ngồi, trọng lượng, tuyến xe...
- Hành động: khởi động xe, chạy xe, dừng xe, tắt máy.

```
class Bus
{
   private:
         char colour[10];
         int seats;
         float weight;
         char ways[30];
   public:
         void start();
         void run();
         void stop();
         void shutdown();
};
```

Câu 1.6: Từ hai lớp Car và Bus của bài 2 và bài 3. Đề xuất một lớp Động cơ (Engine) cho hai lớp trên kế thừa, để tránh trùng lặp dữ liệu giữa hai lớp Car và Bus.

```
class Engine
{
public:
      char colour[10];
      float weight;
```

```
public:
        void start();
        void run();
        void stop();
        void shutdown();
};
```

CHƯƠNG 2: LỚP (CLASS)

2.1 Tóm tắt lý thuyết

- Định nghĩa lớp

Cú pháp: `class tên_lớp`

```
{
 private: [Khai báo các thuộc tính] //có thể thay private
        [Định nghĩa các hàm thành phần (phương thức)]
 public:    [Khai báo các thuộc tính]
            [Định nghĩa các hàm thành phần (phương thức)]
};
```

- Định nghĩa các hàm thành phần:

```
Kiểu_trả_về_của_hàm  Tên_lớp::Tên_hàm(khai báo tham số)
                     { [nội dung hàm]
                       }
```

- Tạo lập đối tượng

Sau khi định nghĩa lớp, ta có thể khai báo các biến thuộc kiểu lớp. Các biến này được gọi là các đối tượng. Cú pháp khai báo biến đối tượng như sau:

```
Tên_lớp    Danh_sách_biến ;
```

Đối tượng cũng có thể khai báo khi định nghĩa lớp theo cú pháp sau:

```
class tên_lớp
      {
         ...
      } <Danh_sách_biến>;
```

2.2 Các dạng bài tập

<u>Câu 2.1:</u> Cho biết kết quả khi thực hiện chương trình sau:

```
#include <iostream.h>
class          samp   {
  int      a, b;
public:
  samp(int   n, int       m)     {       a=n; b=m;     }
  int      get_a()                {return  a;     }
  int      get_b()                {return  b;     }
};
void           main() {
   samp ob[4]= {samp(1, 2), samp(3, 4), samp(5, 6), samp(7, 8)};
   int     i;
   samp *p;
   p=ob;
   for(i=0; i<4; i++)
   { cout <<p->get_a()<< " "; cout <<p->get_b()<< " "; p++; }
   cout <<"\n";
}
```

Hướng dẫn:

Với dạng bài này cần nắm vững ý nghĩa, tác dụng của từng câu lệnh trong chương trình để từ đó xác định được kết quả hiển thị ra màn hình khi chương trình được thực hiện.

Trong chương trình trên khi ta truyền tham số samp(1,2) khi đó tham số n, m trong samp sẽ lần lượt nhận giá trị là 1 và 2. Sau đó a sẽ nhận giá trị của n tức là 1, b sẽ nhận giá trị của m tức là 2 để trả lại kết quả cho hai giá trị a, b do đó trên màn hình sẽ in ra giá trị của a và b lần lượt là 1 và 2, tương tự cho các samp tiếp theo. Sau khi chương trình được thực hiện kết quả hiển thị ra màn hình lần lượt là:

1 2 3 4 5 6 7 8

2.3 Các vấn đề về thảo luận, thực hành, thí nghiệm

Chương này cung cấp kiến thức cơ bản về lập trình hướng đối tượng C++ cho sinh viên, cơ chế hoạt động ý nghĩa của các câu lệnh đơn giản về lập trình hướng đối tượng từ đó giúp sinh viên bước đầu làm quen với lập trình hướng đối tượng. Kết thúc chương 2 yêu cầu sinh viên phải hiểu và viết được những chương trình đơn giản

2.4 Bài tập sinh viên tự làm

Câu 2.2: Cho biết kết quả khi thực hiện chương trình sau:

```
#include <iostream.h>
class samp {
 int a;
public:
 void set_a(int n) { a=n; }
 int get_a() {return a; } };
void main() {
samp ob[4]; int i;
for(i=0; i<4; i++) ob[i].set_a(i);
for(i=0; i<4; i++) cout<< ob[i].get_a();
```

```
cout << "\n"; }
```

Sau khi chương trình được thực hiện kết quả hiển thị ra màn hình lần lượt là:

```
0123
```

Câu 2.3: Cho biết kết quả khi thực hiện chương trình sau:

```
#include <iostream.h>
class samp {
  int i;
public:
  samp(int n) {i=n; }
  int get_i() {return i; }
};
int sqr_it(samp o) {return o.get_i() * o.get_i(); }
void main() {
    samp a(10), b(2);
    cout << sqr_it(a) <<"\n";
    cout << sqr_it(b) << "\n";
}
```

Sau khi chương trình được thực hiện kết quả hiển thị ra màn hình lần lượt là:

```
100
4
```

Câu 2.4: Cho biết kết quả khi thực hiện chương trình sau:

```
#include <iostream.h>
class samp {
  int i;
```

```
public:
  samp(int      n)
        {i=n;    }
  int     get_i()
        {return          i;       }
};
int       sqr_it(samp   o){     return  o.get_i()*o.get_i();   }
void            main() {
samp     a(10), b(4);
cout << sqr_it(a) <<"\n";
cout << sqr_it(b) << "\n";
}
```

Sau khi chương trình được thực hiện kết quả hiển thị ra màn hình lần lượt là:

100

16

<u>Câu 2.5:</u> Viết chương trình xây dựng một lớp a tính giá trị của tổng sau:

S=1+2+3+...+n (n nguyên dương)

```
#include<iostream.h>
class a{
      int n,i;
      float s;
public:
      void nhap(){
            cout<<"nhap n="; cin>>n;}
      float tong(){
            s=0;
```

```
            for(i=1;i<=n;i++) s=s+i;
            return s;}
        void hienthi(){
            cout<<" tong s="<<s;} };
    void main(){
        a th;
        th.nhap();
        th.tong();
        th.hienthi();}
```

Câu 2.6: Xây dựng một lớp TamGiac để mô tả các đối tượng tam giác bao gồm các hàm thành phần như sau:

```
class TamGiac
{
    private:
        double a, b, c; //Ba cạnh tam giác
    public:
        TamGiac(double aa = 0, double bb = 0, double cc = 0);
    void Nhap(); //Nhập ba cạnh
    void Xuat(); //Xuất thông tin tam giác
    int HopLe(); //Kiểm tra ba cạnh tam giác hợp lệ không?
    void PhanLoai(); //Phân loại tam giác
    double ChuVi(); //Tính chu vi tam giác
    double DienTich(); //Tính diện tích tam giác
};
```

```
#include <iostream.h>
#include <math.h>
#include <conio.h>

class tamgiac
{
 private:
       double a,b,c;
 public:
       tamgiac(double aa=0, double bb=0, double cc=0)
       {
             a=aa;
             b=bb;
             c=cc;
             }
       void nhap()
       {
             cout<<"Nhap 3 canh tam giac \n";
             cout<<"Nhap canh a= "; cin>>a;
             cout<<"Nhap canh b= "; cin>>b;
             cout<<"Nhap canh c= "; cin>>c;
             }
       void xuat()
       {
             cout<<"Thong tin tam giac \n";
             cout<<"Canh a= "<<a<<", Canh b= "<<b<<", Canh c= "<<c;
             }
       int hople()
```

```
{
        if ((a+b>c)&&(b+c>a)&&(c+a>b)&&(a>0)&&(b>0)&&(c>0))
        return 1;
        else return 0;
        }
void phanloai()
{
        if
(((a!=b)&&(a!=c))||((b!=a)&&(b!=c))||((c!=a)&&(c!=b)))
        {
        if ((a==b)||(b==c)||(c==a))
                cout<<"Tam giac can";
                else if
((a*a==b*b+c*c)||(b*b==a*a+c*c)||(c*c==a*a+b*b))
                        cout<<"Tam giac vuong";
        else cout<<"Tam giac thuong";
        }
        else cout<<"Tam giac deu";
}
double chuvi()
{
        double p;
        p=a+b+c;
        return p;
        }
double dientich()
{
        double p,s;
        p=(a+b+c)/2;
```

```
            s=sqrt(p*(p-a)*(p-b)*(p-c));
            return s;
            }
 };
void main()
{
 tamgiac a;
 a.nhap();
 a.xuat();
 cout<<endl;
 if (a.hople()==1)
 {
 cout<<"3 canh tam giac hop le";
 cout<<endl;
 a.phanloai();
 cout<<"\nChu vi tam giac: "<<a.chuvi();
 cout<<"\nDien tich tam giac: "<<a.dientich();
 }
 else cout<<"3 canh tam giac khong hop le";
 getch();
}
```

<u>Câu 2.7:</u> Xây dựng một lớp Mang1c dữ liệu thành phần kiểu số nguyên, các hàm thành phần gồm:

- Hàm khởi tạo mảng
- Hàm in mảng
- Hàm in phần tử lớn nhất, phần tử nhỏ nhất của mảng

```
#include <iostream.h>
#include <conio.h>
class Mang1c
{
      private:
            int a[100];
            int n;
public:
void input()
{
      cout<<"Input n= "; cin>>n;
      for (int i=0; i<n; i++)
      {
      cout<<"a["<<i<<"]= ";
      cin>>a[i];
      }
}
void show()
{
      for (int i=0; i<n; i++)
      cout<<a[i]<<" ";
}
void max_min()
{
      int max, min;
      max=a[0];
      for (int i=1; i<n; i++)
      {
```

```
            if (max<a[i]) max=a[i];
            }
            cout<<"Maximum value: "<<max;
            min=a[0];
            for (int i=1; i<n; i++)
            {
            if (min>a[i]) min=a[i];
            }
            cout<<"\nMinimum value: "<<min;
      }
};
void main()
{
      Mang1c a;
      a.input();
      a.show();
      cout<<endl;
      a.max_min();
      getch();
}
```

<u>Câu 2.8:</u> Xây dựng lớp Date. Dữ liệu thành phần bao gồm ngày, tháng, năm. Các hàm thành phần bao gồm: hàm tạo, hàm truy cập dữ liệu, hàm normalize() để chuẩn hóa dữ liệu nằm trong khoảng quy định của ngày (1 ≤ ngày <daysIn(tháng)), tháng (1 ≤ tháng < 12), năm (năm ≥ 1), hàm daysIn(int) trả về số ngày trong tháng, hàm advance(int y, int m, int d) để tăng ngày hiện lên các năm y, tháng m, ngày d của đối tượng đang tồn tại và một hàm print() để hiển thị dữ liệu.

```
#include <iostream.h>
#include <conio.h>
class date
{
      public:
            int day, month, year;
      public:
            void create()
            {
                  cout<<"Input the value of day: "; cin>>day;
                  cout<<"Input the value of month: "; cin>>month;
                  cout<<"Input the value of year: "; cin>>year;
                  }
            int get_day()
            {
                  return day;
                  }
            int get_month()
            {
                  return month;
                  }
            int get_year()
            {
                  return year;
                  }
            int daysIn(int month)
            {
                  int daysIn;
```

```
        switch(month)
        {
    case 1: case 3: case 5: case 7: case 8: case 10: case 12:
            daysIn=31;
            break;
            case 4: case 6: case 9: case 11:
            daysIn=30;
            break;
            case 2:
                if (year%4==0)
                daysIn=29;
                else daysIn=28;
            break;
            }
        return daysIn;
        }
    void normalize()
    {
        if ((day<1)||(day>daysIn(month)))
        cout<<"Please input the value of day again!";
        if ((month<1)||(month>12))
        cout<<"\nPlease input the value of month again!";
        if (year<1) cout<<"\nPlease input the value of year
again!";
        }
    int advance(int day, int month, int year)
    {
        day=day+1;
        return day;
```

```
                }
            void print()
            {
                cout<<"Day: "<<advance(day,month,year);
                cout<<"\nMonth: "<<month;
                cout<<"\nYear: "<<year;
                }
        };

void main()
{
        date a;
        a.create();
        cout<<endl;
        a.normalize();
        cout<<endl;
        a.print();
        getch();
}
```

<u>Câu 2.9:</u> Xây dựng lớp ma trận có tên là Matrix cho các ma trận, các hàm thành phần bao gồm: hàm tạo mặc định, hàm nhập xuất ma trận, cộng, trừ, nhân hai ma trận.

```
#include <iostream.h>
#include <conio.h>



class matrix
```

```
{
    private:
        int a[50][50], b[50][50];
        int m, n, k;
    public:
        matrix(int a1=0, int b1=0)
        {
        for (int i=0; i<50; i++)
            for (int j=0; j<50; j++)
            {
            a[i][j]=a1;
            b[i][j]=b1;
            }
        }
        void nhap()
        {
            cout<<"Nhap ma tran a \n";
            cout<<"Nhap so hang m= "; cin>>m;
            cout<<"Nhap so cot n= "; cin>>n;
            for (int i=0; i<m; i++)
                for (int j=0; j<n; j++)
                {
                    cout<<"Nhap a["<<i<<"]["<<j<<"]= ";
                    cin>>a[i][j];
                }
            cout<<"\nNhap ma tran b \n";
            cout<<"So hang n= "<<n;
            cout<<"\nNhap so cot k= "; cin>>k;
```

```
        for (int i=0; i<n; i++)
                for (int j=0; j<k; j++)
                {
                        cout<<"Nhap b["<<i<<"]["<<j<<"]= ";
                        cin>>b[i][j];
                }
        }
void xuat()
{
        cout<<"Ma tran a: \n";
        for (int i=0; i<m; i++)
        {
                for (int j=0; j<n; j++)
                {
                cout<<a[i][j]<<"   ";
                }
                cout<<endl;
        }
        cout<<"\nMa tran b: \n";
        for (int i=0; i<n; i++)
        {
                for (int j=0; j<k; j++)
                {
                        cout<<b[i][j]<<"   ";
                }
                cout<<endl;
        }
}
```

```
void cong_tru_nhan()
{
	int c[50][50];
	int d[50][50];
	int e[50][50];
	cout<<"\nCong hai ma tran: \n";
	for (int i=0; i<m; i++)
		for (int j=0; j<n; j++)
		{
			c[i][j]=a[i][j]+b[i][j];
		}
	for (int i=0; i<m; i++)
	{
		for (int j=0; j<n; j++)
		{
			cout<<c[i][j]<<"  ";
		}
		cout<<endl;
	}
	cout<<"\nTru hai ma tran: \n";
	for (int i=0; i<m; i++)
		for (int j=0; j<n; j++)
		{
			d[i][j]=a[i][j]-b[i][j];
		}
	for (int i=0; i<m; i++)
	{
		for (int j=0; j<n; j++)
```

```
                {
                        cout<<d[i][j]<<"  ";
                }
                cout<<endl;
        }
        cout<<"\nNhan hai ma tran: \n";
        //nhan hai ma tran
        for (int i=0; i<m; i++)
                for (int j=0; j<k; j++)
                {
                        e[i][j]=0;
                        for (int d=0; d<n; d++)
                        {
                                e[i][j]=e[i][j]+a[i][d]*b[d][j];
                        }
                }
        for (int i=0; i<m; i++)
        {
                for (int j=0; j<k; j++)
                {
                        cout<<e[i][j]<<"  ";
                }
                cout<<endl;
        }
    }
};
```

```
void main()
{
      matrix a;
      a.nhap();
      cout<<endl;
      a.xuat();
      a.cong_tru_nhan();
      getch();
}
```

<u>Câu 2.10:</u> Xây dựng một lớp Vector để mô tả các đối tượng vector trong không gian n chiều và thực hiện các công việc sau: nhập tọa độ, xuất tọa độ, cộng, trừ hai vector, nhân vô hướng hai vector.

```
#include <iostream.h>
#include <conio.h>

class vector
{
 private:
      int x, y;
 public:
      void nhap()
      {
            cout<<"Nhap x= "; cin>>x;
            cout<<"Nhap y= "; cin>>y;
            cout<<endl;
            }

      void xuat()
```

```
        {
            cout<<"("<<x<<","<<y<<")";
            cout<<endl;
            }
        void cong_tru_nhan(vector a, vector b)
        {
            vector c,d,e;
            c.x=a.x+b.x;
            c.y=a.y+b.y;
            cout<<"\nTong hai vector: ";
            c.xuat();
            d.x=a.x-b.x;
            d.y=a.y-b.y;
            cout<<"\nHieu hai vector: ";
            d.xuat();
            e.x=a.x*b.x+a.x*b.y;
            e.y=a.y*b.x+a.y*b.y;
            cout<<"\nTich vo huong hai vector: ";
            e.xuat();
        }
    };

void main()
{
 vector a,b,c;
 cout<<"Nhap vector thu nhat \n";
 a.nhap();
 cout<<"Nhap vector thu hai \n";
```

```
        b.nhap();

        c.cong_tru_nhan(a,b);

        getch();

       }
```

<u>Câu 2.11:</u> Nhập 4 phân số từ bàn phím. Viết chương trình có lớp là PS để xác định phân số lớn nhất.

```
#include <iostream.h>
#include <conio.h>
#include <math.h>
#include <iomanip.h>

class ps
{
      public:
            int tu, mau;
      public:
            void input()
            {
                  cout<<"Nhap tu: "; cin>>tu;
                  cout<<"Nhap mau: "; cin>>mau;
                  }
            void display()
            {
                  cout<<tu<<"/"<<mau;
                  }
            int ucln(int a, int b)
            {
                  int x=abs(a);
                  int y=abs(b);
                  if (x*y==0) return 1;
                  while (x!=y)
                  {
                        if (x>y) x=x-y;
                        else y=y-x;
                        }
                  return x;
                  }
      };


void main()
{
      ps a[5];
      for (int i=0; i<4; i++)
      {
```

```
            do
            {cout<<"Nhap phan so thu "<<i+1<<": \n";
            a[i].input();
            }
            while (a[i].mau==0);
            cout<<endl;
            }
      cout<<"=>Cac phan so da nhap: ";
      for (int i=0; i<4; i++)
      {
            a[i].display();
            cout<<setw(3);
            }
      int c,b;
      ps max=a[0];
      for (int i=1; i<4; i++)
      {
            c=max.tu*a[i].mau;
            b=max.mau*a[i].tu;
            if (c<b)
            {
                  max=a[i];
                  }
            }
      cout<<endl;
      cout<<"\n=>Phan so lon nhat: ";
      max.display();
      getch();
}
```

<u>Câu 2.12:</u> Viết chương trình xây dựng một lớp có tên là time để nhập giờ : phút : giây bất kỳ từ bàn phím. Hãy xác định số giờ, phút, giây lớn nhất trong các tham số bất kỳ từ bàn phím

```
#include <iostream.h>
#include <conio.h>
#include <iomanip.h>

class time
{
      public:
            int gio, phut, giay;
      time(int gio1=0, int phut1=0, int giay1=0)
      {
            gio=gio1;
            phut=phut1;
            giay=giay1;
            }
      public:
            void nhap()
```

```
            {
                  cout<<"-Nhap gio: "; cin>>gio;
                  cout<<"-Nhap phut: "; cin>>phut;
                  cout<<"-Nhap giay: "; cin>>giay;
                  }
            void hienthi()
            {
                  cout<<gio<<":"<<phut<<":"<<giay;
                  }
            int sec()
            {
                  return gio*3600+phut*60+giay;
                  }
      };


void main()
{
      time a[10];
      int n;
      cout<<"Nhap so bo thoi gian n= ";
      cin>>n;
      for (int i=0; i<n; i++)
      {
            cout<<"Nhap bo thoi gian thu "<<i+1<<" \n";
            a[i].nhap();
            cout<<endl;
            }
      cout<<"\nBo thoi gian da nhap: \n";
      for (int i=0; i<n; i++)
      {
            a[i].hienthi();
            cout<<setw(3);
            }
      cout<<endl;
      time max=a[0];
      for (int i=1; i<n; i++)
      {
            if (max.sec()<a[i].sec())
            {
            max=a[i];
            }
      }
      cout<<"\nThoi gian max: ";
      max.hienthi();
      getch();
}
```

CHƯƠNG 3: TOÁN TỬ TẢI BỘI

3.1 Tóm tắt lý thuyết

- Định nghĩa toán tử tải bội:
  Các toán tử cùng tên thực hiện nhiều chức năng khác nhau được gọi là toán tử tải bội. Dạng định nghĩa tổng quát của toán tử tải bội như sau:

```
Kiểu_trả_về   operator   op(danh sách tham số)
              {//thân toán tử}
```

Trong đó: Kiểu_trả_về là kiểu kết quả thực hiện của toán tử.

op là tên toán tử tải bội

operator op(danh sách tham số) gọi là hàm toán tử tải bội, nó có thể là hàm thành phần hoặc là hàm bạn, nhưng không thể là hàm tĩnh.

- Định nghĩa chồng các toán tử ++ , --

  Ta có thể định nghĩa chồng cho các toán tử ++/-- theo quy định sau:

- Toán tử ++/-- dạng tiền tố trả về một tham chiếu đến đối tượng thuộc lớp.
- Toán tử ++/-- dạng tiền tố trả về một đối tượng thuộc lớp.

3.2 Các dạng bài tập

<u>Câu 3.1:</u> Xây dựng một lớp Diem gồm các thuộc tính M(x,y,z) là tọa độ của một điểm bất kỳ. Xây dựng toán tử nạp chồng + để tính tọa độ của điểm M bất kỳ trên hệ tọa độ O(x,y,z) biết rằng tọa độ của M bằng tổng tọa độ của hai điểm M1, M2

Hướng dẫn:

```
#include <iostream.h>
#include <conio.h>

class Diem
   {
     private:
       float x,y,z;
     public:
       Diem() {}
       Diem(float x1,float y1,float z1)
      { x = x1; y = y1; z=z1;}
       friend Diem  operator +(Diem d1, Diem d2)
           {  Diem tam;
              tam.x = d1.x + d2.x;
              tam.y = d1.y + d2.y;
              tam.z = d1.z + d2.z;
              return tam;
           }
       void hienthi()
           { cout<<x<<" "<<y <<" " <<z<<endl;}
   };
void main() {
     clrscr();
     Diem d1(3,-6,8),d2(4,3,7),d3;
     d3=d1+d2;
     d1.hienthi();
     d2.hienthi();
```

```
        cout<<"\n Tong hai diem co toa do la :";
        d3.hienthi();
        getch();
    }
```

3.3 Các vấn đề về thảo luận, thực hành, thí nghiệm

Chương này cung cấp cho sinh viên biết thế nào là toán tử tải bội, cách nạp chồng toán tử từ đó biết vận dụng làm một số bài tập về toán tử tại bội. Sau khi kết thúc chương 3 yêu câu sinh viên phải thực hiện việc làm chồng một số toán tử cơ bản bằng cách thực hiện các bài tập tổng hợp phía dưới đây

3.4 Bài tập sinh viên tự làm

Câu 3.2: Viết chương trình thực hiện các công việc sau:

- Tạo một lớp Point với dữ liệu kiểu nguyên để biểu diễn toạ độ của một điểm.
- Xây dựng một constructor để tạo đối tượng, constructor sử dụng các tham số có giá trị ngầm định.
- Xây dựng hàm thành phần public display() để thực hiện in điểm ra màn hình.
- Xây dựng hàm di chuyển điểm đến một toạ độ mới.
- Xây dựng chồng toán tử operator - để lấy điểm đối xứng qua gốc toạ độ.

Xây dựng hàm main để kiểm tra lớp đã tạo.

```
#include <iostream.h>
#include <conio.h>

class point
{
    private:
        int x,y;
```

```
    public:
        point(int x1=1, int y1=2)
        {
            x=x1;
            y=y1;
            }
        void display()
        {
            cout<<"Toa do: ("<<x<<","<<y<<")";
            }
        void tinhtien()
        {
            float delta_x, delta_y;
            cout<<"Nhap khoang tinh tien hoanh do x= ";
              cin>>delta_x;
            cout<<"Nhap khoang tinh tien tung do y= ";
              cin>>delta_y;
        cout<<"Toa do diem moi: ("<<x+delta_x<<","<<y+delta_y<<")";
        }
        point operator-(point a)
        {
            a.x=-1*a.x;
            a.y=-1*a.y;
            return a;
        }
    };
void main()
{
```

```
        point a,b;
        a.display();
        cout<<endl;
        a.tinhtien();
        cout<<endl;
        b=b-a;
        cout<<"Diem doi xung ";
        b.display();
        getch();
}
```

Câu 3.3: Tạo một lớp PS để thực hiện các thao tác số học với phân số trong đó:

- Sử dụng các biến nguyên để biểu diễn các thành phần dữ liệu của tử số và mẫu số của lớp.
- Xây dựng một constructor để tạo đối tượng, constructor sử dụng các tham số có giá trị ngầm định.
- Xây dựng hàm thành phần public để thực hiện in phân số ra màn hình dưới dạng a/b trong đó a là tử số, b là mẫu số; hàm tối giản một phân số.
- Thực hiện chồng toán tử operator cho các thao tác: Cộng, trừ, nhân, chia hai phân số.

Xây dựng hàm main để kiểm tra lớp đã tạo.

```
#include <iostream.h>
#include <math.h>
#include <conio.h>


class ps
{
```

```
public:
      int t,m;
ps(int t1=0, int m1=1)
{
      t=t1;
      m=m1;
      }
public:
      void input()
      {
            cout<<"Nhap tu so: "; cin>>t;
            cout<<"Nhap mau so: "; cin>>m;
            }
      int ucln(int a, int b)
      {
            int x,y;
            x=abs(a); y=abs(b);
            if (x*y==0) return 1;
            while (x!=y)
            {
                  if (x>y) x=x-y;
                  else y=y-x;
                  }
                  return x;
            }
      friend ps operator+(ps a, ps b)
      {
            ps c;
```

```
        c.t=a.t*b.m+b.t*a.m;
        c.m=a.m*b.m;
        return c;
        }
friend ps operator-(ps a, ps b)
{
        ps c;
        c.t=a.t*b.m-b.t*a.m;
        c.m=a.m*b.m;
        return c;
        }
friend ps operator*(ps a, ps b)
{
        ps c;
        c.t=a.t*b.t;
        c.m=a.m*b.m;
        return c;
        }
friend ps operator/(ps a, ps b)
{
        ps c;
        c.t=a.t*b.m;
        c.m=a.m*b.t;
        return c;
        }
void display()
{
        cout<<t/ucln(t,m)<<"/"<<m/ucln(t,m);
```

```
                    }
};



void main()
{
      ps a,b,c,d,e,f;
      cout<<"NHAP CAC PHAN SO: \n";
      cout<<"-----------------------\n";
      cout<<"Nhap phan so thu nhat: \n";
      a.input();
      cout<<"\nNhap phan so thu hai: \n";
      b.input();
      c=a+b; d=a-b; e=a*b; f=a/b;
      cout<<"\nCAC PHEP TINH TOAN HAI PHAN SO: \n";
      cout<<"\nTong: ";
      c.display();
      cout<<"\nHieu: ";
      d.display();
      cout<<"\nTich: ";
      e.display();
      cout<<"\nThuong: ";
      f.display();
      getch();
}
```

Câu 3.4: Lập chương trình thực hiện các công việc sau:

- Tạo một lớp Complex để thực hiện các thao tác số học với các số phức trong đó:

  Số phức có dạng : <Phần thực> + <Phần ảo> *j

- Sử dụng các biến thực để biểu diễn các thành phần dữ liệu riêng của lớp.

- Xây dựng một constructor để tạo đối tượng, constructor sử dụng các tham số có giá trị ngầm định.

- Xây dựng hàm thành phần public để thực hiện in số phức ra màn hình dưới dạng (a, b) trong đó a là phần thực, b là phần ảo.

- Thực hiện chồng toán tử operator cho các thao tác : Cộng, trừ, nhân, chia hai số phức.

Xây dựng hàm main để kiểm tra lớp đã tạo.

```
#include <iostream.h>
#include <math.h>
#include <conio.h>
#include <iomanip.h>

class complex
{
      public:
            float thuc, ao;
      complex(float thuc1=0, float ao1=0)
      {
      thuc=thuc1;
      ao=ao1;
      }
      public:
```

```
void input()
{
      cout<<"Nhap phan thuc: "; cin>>thuc;
      cout<<"Nhap phan ao: "; cin>>ao;
      }
friend complex operator+(complex a, complex b)
{
      complex c;
      c.thuc=a.thuc+b.thuc;
      c.ao=a.ao+b.ao;
      return c;
}
friend complex operator-(complex a, complex b)
{
      complex c;
      c.thuc=a.thuc-b.thuc;
      c.ao=a.ao-b.ao;
      return c;
}
friend complex operator*(complex a, complex b)
{
      complex c;
      c.thuc=a.thuc*b.thuc-a.ao*b.ao;
      c.ao=a.thuc*b.ao+a.ao*b.thuc;
      return c;
}
friend complex operator/(complex a, complex b)
{
```

```
            complex c;
            float tongbp;
            tongbp=pow(b.thuc,2)+pow(b.ao,2);
            c.thuc=(a.thuc*b.thuc-a.ao*b.ao)/tongbp;
            c.ao=(a.thuc*b.ao+a.ao*b.thuc)/tongbp;
            return c;
      }
      void display()
      {
            cout<<"("<<thuc<<","<<ao<<")";
            cout<<endl;
            }
};


void main()
{
      complex a,b,c,d,e,f;
      cout<<"Nhap so phuc thu nhat: \n";
      a.input();
      cout<<endl;
      cout<<"\nNhap so phuc thu hai: \n";
      b.input();
      cout<<endl;
      c=a+b;
      d=a-b;
      e=a*b;
      f=a/b;
```

```
        cout<<"Tong: ";
        c.display();
        cout<<"Hieu: ";
        d.display();
        cout<<"Tich: ";
        e.display();
        cout<<"Thuong: ";
        cout<<setprecision(2);
        f.display();
        getch();
}
```

Câu 3.5: Viết chương trình xây dựng một lớp có tên là DATA bao gồm:

- Dữ liệu: a,b là hai số bất kỳ nhập từ bàn phím
- Phương thức: Nhập, xuất dữ liệu, hàm thành phần tìm ước chung lớn nhất của hai số đó.

Xây dựng hàm main() để kiểm tra

```
#include <iostream.h>
#include <conio.h>

class DATA
{
        private:
                int a,b;
        public:
                void nhap()
                {
                        cout<<"Nhap a= "; cin>>a;
                        cout<<"Nhap b= "; cin>>b;
                        }
                void xuat()
                {
                        cout<<"a= "<<a<<", b= "<<b<<" ";
                        }
                int get_a()
                {
                        return a;
                        }
                int get_b()
                {
```

```
                return b;
                }
            int ucln(int a, int b)
            {
                if (a==0||b==0) return a+b;
                if (a==b) return a;
                if (a>b) return ucln(a-b,b);
                else return ucln(a,b-a);
                }
        };

void main()
{
        DATA x;
        x.nhap();
        x.xuat();
        cout<<"Uoc chung lon nhat: "<<x.ucln(x.get_a(),x.get_b());
        getch();
}
```

<u>Câu 3.6:</u> Tạo một lớp S1 để thực hiện các thao tác số học với hai số bất kỳ nhập từ bàn phím. Thực hiện chồng toán tử operator cho các thao tác: Cộng, trừ, nhân, chia đối với hai số đó.

Xây dựng hàm main() để kiểm tra lớp đã tạo.

```
#include <iostream.h>
#include <conio.h>

class S1
{
      private:
            float x;
      public:
            void nhap()
            {
                  cout<<"Nhap so = "; cin>>x;
                  }
            void xuat()
            {
                  cout<<"So = "<<x;
                  cout<<endl;
                  }
            friend S1 operator+(S1 a, S1 b)
            {
                  S1 c;
                  c.x=a.x+b.x;
                  return c;
                  }
            friend S1 operator-(S1 a, S1 b)
```

```
        {
            S1 c;
            c.x=a.x-b.x;
            return c;
            }
        friend S1 operator*(S1 a, S1 b)
        {
            S1 c;
            c.x=a.x*b.x;
            return c;
            }
        friend S1 operator/(S1 a, S1 b)
        {
            S1 c;
            c.x=a.x/b.x;
            return c;
            }
    };

void main()
{
    S1 a,b,c,d,e,f;
    a.nhap();
    b.nhap();
    c=a+b; d=a-b; e=a*b; f=a/b;
    c.xuat(); d.xuat(); e.xuat(); f.xuat();
    getch();
}
```

CHƯƠNG 4: KẾ THỪA

4.1 Tóm tắt lý thuyết

- Đơn **kế thừa**

- Định nghĩa lớp dẫn xuất từ một lớp cơ sở

Giả sử đã định nghĩa lớp A. Cú pháp để xây dựng lớp B dẫn xuất từ lớp A như sau:

```
class  B: mode A
{
        private:
                // Khai báo các thuộc tính lớp B
        public:
            // Định nghĩa các hàm thành phần lớp B
};
```

Trong đó mode có thể là **private** hoặc **public** với ý nghĩa như sau:

- Kế thừa theo kiểu public thì tất cả các thành phần public của lớp cơ sở cũng là thành phần public của lớp dẫn xuất.

- Kế thừa theo kiểu private thì tất cả các thành phần public của lớp cơ sở sẽ trở thành các thành phần private của lớp dẫn xuất.

- **Đa kế thừa**

- Định nghĩa lớp dẫn xuất từ nhiều lớp cơ sở

  Giả sử đã định nghĩa các lớp A, B. Cú pháp để xây dựng lớp C dẫn xuất từ lớp A và B như sau:

```
class  C: mode A, mode B
      {
          private:
                      // Khai báo thuộc tính
          public:
          // Các hàm thành phần
};
```

trong đó mode có thể là private, public hoặc protected. Ý nghĩa của kiểu dẫn xuất này giống như trường hợp đơn kế thừa.

4.2 Các dạng bài tập

Câu 4.1: Viết chương trình giải hệ phương trình bậc nhất hai ẩn bằng cách sử dụng đơn kế thừa:

Hướng dẫn:

```
#include <iostream.h>
class hpt {
 public:
   float a1,a2,b1,b2,c1,c2,x,y;
   void nhap () {
        cout<<"Nhap he so a1=";cin>>a1;
        cout<<"Nhap he so b1=";cin>>b1;
        cout<<"Nhap he so c1=";cin>>c1;
        cout<<"Nhap he so a2=";cin>>a2;
        cout<<"Nhap he so b2=";cin>>b2;
        cout<<"Nhap he so c2=";cin>>c2;
        }
   void xuat (){
```

```
        cout<<"He phuong trinh co nghiem la"<<endl;
         cout<<"x="<<x<<endl;
         cout<<"y="<<y; }
         };
class ghpt: public hpt
 { public:
 float D,Dx,Dy;
 float nghiem (){
  D=a1*b2-b1*a2;
  Dx= b1*c2 - b2*c1;
  Dy= a1*c2-a2*c1;
 if(D!=0) {x = Dx/D;
        y=Dy/D;}
 else cout<<"hpt vo nghiem";
 return x,y;}
 };
 void main () {
  ghpt pt;
  pt.nhap();
  pt.nghiem();
  pt.xuat(); }
```

<u>Câu 4.2:</u> Viết chương trình minh họa việc quản lý kết quả thi của một lớp không quá 100 sinh viên. Chương trình gồm 3 lớp: lớp cơ sở sinh viên (sinhvien) chỉ lưu họ tên và số báo danh, lớp điểm thi (diemthi) kế thừa lớp sinh viên và lưu kết quả môn thi 1 và môn thi 2. Lớp kết quả (ketqua) lưu tổng số điểm đạt được của sinh viên.

```
#include <iostream.h>
#include <conio.h>
#include <stdio.h>

class sinhvien
{ char hoten[25];
  protected:
      int sbd;
  public:
      void nhap()
        {   cout<<"\nHo ten: "; fflush(stdin); gets(hoten);
            cout<<"So bao danh: "; cin>>sbd;
          }
      void hienthi()
        { cout<<"So bao danh: "<<sbd<<endl;
          cout<<"Ho va ten sinh vien: "<<hoten<<endl;
         }
};
class diemthi: public sinhvien
{ protected :
      float d1,d2;
  public :
      void nhap_diem()
      {
       cout<<"Nhap diem hai mon thi: \n";
       cin>>d1>>d2;
      }
      void hienthi_diem()
```

```
        {  cout<<"Diem mon 1: "<<d1<<endl;
           cout<<"Diem mon 2: "<<d2<<endl;
        }
};
class ketqua: public diemthi
{
  float tong;
  public:
      void display()
      { tong = d1+d2;
       hienthi();
       hienthi_diem();
       cout<<"Tong so diem: "<<tong<<endl;
     }
};
void main()
{ int i,n; ketqua sv[100];
  cout<<"\n Nhap so sinh vien: ";
  cin>>n;
  //clrscr();
  for(i=0;i<n;++i)
  { sv[i].nhap();
    sv[i].nhap_diem();
  }
  for(i=0;i<n;++i)
    sv[i].display();
  getch();
}
```

4.3 Các vấn đề về thảo luận, thực hành, thí nghiệm

Chương này cung cấp kiến thức cơ bản nhất của lập trình hướng đối tượng là cách sử dụng phương pháp kế thừa(đơn kế thừa, đa kế thừa) trong lập trình hướng đối tượng, đây là sự khác biệt cơ bản giữa lập trình hướng đối tượng so với các phương pháp lập trình truyền thống. Chương 4 cũng thể hiện được tính linh hoạt tiên dụng của phương pháp lập trình hướng đối tượng. Kết thúc chương 4 yêu cầu sinh viên nắm rõ về cách thức sử dụng kế thừa trong việc giải quyết các bài toán phức tạp bằng cách thực hiện các bài tập tổng hợp dưới đây

4.4 Bài tập sinh viên tự làm

<u>Câu 4.3:</u> Lập chương trình thực hiện các công việc sau:

- Xây dựng lớp cơ sở bệnh nhân gồm:
  + Thuộc tính: họ tên, quê quán, năm sinh
  + Phương thức: Nhập, xuất thông tin
- Xây dựng lớp bệnh án kế thừa từ lớp bệnh nhân có thêm:
  + Thuộc tính: tên bệnh án, số tiền viện phí
  + Phương thức: Nhập, xuất thông tin, tính tuổi hiện tại
- Chương trình chính thực hiện:
  + Nhập danh sách N bệnh án
  + Sắp xếp danh sách theo tuổi giảm dần của các bệnh nhân
  + Hiện ra màn hình danh sách các bệnh nhân tuổi<=10.
- Cho biết thông tin các bệnh nhân có tiền viện phí cao nhất

```
#include <iostream.h>
#include <stdio.h>
#include <conio.h>

class benhnhan
{
	public:
		char hoten[25];
		char quequan[30];
		int namsinh;
	public:
		void nhap()
		{
			cout<<"Nhap ten: "; fflush(stdin); gets(hoten);
			cout<<"Nhap que quan: "; fflush(stdin); gets(quequan);
			cout<<"Nhap nam sinh: "; cin>>namsinh;
			}
		void xuat()
		{
			cout<<"\nTen: "<<hoten;
			cout<<"\nQue quan: "<<quequan;
			cout<<"\nNam sinh: "<<namsinh;
			}
	};
class benhan: public benhnhan
{
	public:
		char tenba[30];
```

```
            double tienvp;
        int tuoi()
        {
            int tuoi=2015-namsinh;  //year at the moment - year of birth
            return tuoi;
            }
        void nhap()
        {
            benhnhan::nhap();
            cout<<"Nhap ten benh an: "; fflush(stdin); gets(tenba);
            cout<<"Nhap tien vien phi: "; cin>>tienvp;
            cout<<endl;
            }
        void xuat()
        {
            benhnhan::xuat();
            cout<<"\nTen benh an: "<<tenba;
            cout<<"\nTien vien phi: "<<tienvp;
            cout<<endl;
            }
};

void main() {
        benhan a[50];
        int n,i;
        do
        {
        cout<<"Nhap so benh an n= ";
```

```
cin>>n;
}
while (n<0);
for (i=0; i<n; i++)
{
      cout<<"Nhap thong tin thu "<<i+1<<" \n";
      a[i].nhap();
      }
cout<<"\n------------\n";
for (i=0; i<n; i++)
{
      cout<<"\n-Benh nhan thu : "<<i+1<<" \n";
      a[i].xuat();
      }
for (i=0; i<n-1; i++)
      for (int j=i+1; j<n; j++)
      {
            if (a[j].tuoi()>a[i].tuoi())
            {
                  benhan c=a[i];
                  a[i]=a[j];
                  a[j]=c;
                  }
            }
cout<<"\n------------\n";
cout<<"\nThong tin benh an giam dan ve tuoi: \n";
for (i=0; i<n; i++)
{
```

```
        a[i].xuat();
        }
    cout<<"\n------------\n";
    cout<<"\nDanh sach benh nhan duoi 10 tuoi: ";
    for (i=0; i<n; i++)
    {
        if (a[i].tuoi()<=10)
        a[i].xuat();
        }
    benhan vpmax=a[0];
    for (i=0; i<n; i++)
    {
        if (vpmax.tienvp<a[i].tienvp)
        {
        vpmax=a[i];
        }
    }
    cout<<"\n------------\n";
    cout<<"\nThong tin benh nhan co vien phi cao nhat: ";
    vpmax.xuat();
    getch();
}
```

Câu 4.4: Lập chương trình thực hiện các công việc sau:

- Xây dựng lớp cơ sở sản phẩm gồm:
  + Thuộc tính: Tên sản phẩm, năm sản xuất, giá thành
  + Phương thức: Nhập, xuất thông tin
- Xây dựng lớp hoá đơn bán sản phẩm kế thừa từ lớp sản phẩm có thêm:
  + Thuộc tính: Số lượng bán, giá bán
  + Phương thức: Nhập, xuất thông tin, tính thành tiền (=số lượng * giá bán), tính thuế (=10% thành tiền), tính lãi (chênh lệch giá * số lượng bán)
- Chương trình chính thực hiện:
  + Nhập danh sách N hoá đơn bán sản phẩm
  + Sắp xếp danh sách theo tiền lãi giảm dần
  + Hiện ra màn hình danh sách gồm: số thứ tự, tên sản phẩm, giá thành, số lượng bán, giá bán, thành tiền, thuế và tiền lãi.
  + Tính tổng tiền của các hoá đơn bán sản phẩm

Cho biết thông tin các hoá đơn bán sản phẩm có tiền thuế cao nhất

```
#include <iostream.h>
#include <stdio.h>
#include <conio.h>

class sanpham
{
 public:
      char tensp[25];
      int namsx;
      float gia;
 public:
```

```
        void nhap()
        {
              cout<<"Nhap ten san pham: "; fflush(stdin);
              gets(tensp);
              cout<<"Nhap nam san xuat: "; cin>>namsx;
              cout<<"Nhap gia: "; cin>>gia;
        }
        void xuat()
        {
              cout<<"\nTen SP: "<<tensp;
              cout<<"\nNam SX: "<<namsx;
              cout<<"\nGia thanh: "<<gia;
        }
  };

class hoadon: public sanpham
{
 public:
  int slb;
  float giab;
 public:
        void nhap()
        {
              sanpham::nhap();
              cout<<"Nhap so luong ban: "; cin>>slb;
              cout<<"Nhap gia ban: "; cin>>giab;
              cout<<endl;
              }
```

```
        void xuat()
        {
              sanpham::xuat();
              cout<<"\nSo luong ban: "<<slb;
              cout<<"\nGia ban: "<<giab;
              cout<<endl;
              }
        float thanhtien()
        {
              float tt;
              tt=slb*giab;
              return tt;
              }
        float thue()
        {
              float thue;
              thue=(10*thanhtien())/100;
              return thue;
              }
        float lai()
        {
              float lai;
              lai=(giab-gia)*slb;
              return lai;
              }
  };
void main()
{
```

```
hoadon a[100];
int n,i;
cout<<"Nhap so hoa don n= "; cin>>n;
cout<<"\n---------------\n";
for (i=0; i<n; i++)
{
      cout<<"\nNhap hoa don thu "<<i+1<<" \n";
      a[i].nhap();
      }
cout<<"\n---------------\n";
for (i=0; i<n; i++)
{
cout<<"\nThong tin hoa don thu "<<i+1<<" \n";
a[i].xuat();
}
for (i=0; i<n-1; i++)
      for (int j=i+1; j<n; j++)
      {
            if (a[j].lai()>a[i].lai())
            {
                  hoadon c=a[i];
                  a[i]=a[j];
                  a[j]=c;
                  }
            }
cout<<"\n---------------\n";
cout<<"\nDanh sach hoa don giam dan ve lai: \n";
for (i=0; i<n; i++)
```

```
 {
        a[i].xuat();
        }
 cout<<"\n---------------\n";
 cout<<"\nThong tin chi tiet cua san pham: \n";
 for (i=0; i<n; i++)
 {
        cout<<"\nSTT: "<<i+1<<" ";
        a[i].xuat();
        cout<<"Thanh tien: "<<a[i].thanhtien();
        cout<<"\nTien thue: "<<a[i].thue();
        cout<<"\nTien lai: "<<a[i].lai();
        cout<<endl;
        }
 float tongtien=0;
 for (i=0; i<n; i++)
 {
        tongtien+=a[i].thanhtien();
        }
 cout<<"\n---------------\n";
 cout<<"\nTong tien cua cac hoa don ban hang: "<<tongtien;
getch();
}
```

<u>Câu 4.5:</u> Trong một trường Đại học cần quản lý cán bộ và giảng viên. Cán bộ gồm các thông tin: Mã cán bộ, mã đơn vị, năm sinh, hệ số lương, phụ cấp ăn ca, bảo hiểm. Giảng viên cần bổ sung thêm các thông tin: phụ cấp đứng lớp =25% lương cơ bản và phụ cấp độc hại (nếu có) = 10% lương cơ bản.

- Hãy lập chương trình quản lý cán bộ và giảng viên sao cho kế thừa được các dữ liệu dùng chung thực hiện tính lương cho cán bộ và giảng viên tương ứng, in ra danh sách cán bộ và giảng viên phải đóng thuế thu nhập (lương thực lĩnh>5000000).

```
#include <iostream.h>
#include <conio.h>
#include <stdio.h>

float luongcb_cb; //luong co ban cua can bo (trieu dong)
float luongcb_gv; //luong co ban cua giang vien (trieu dong)

class canbo
{
      public:
            char macb[15];
            char madv[15];
            int ns;
            float hsl;
            float pc_anca;
            char bh[20];

      public:
            void nhap()
            {
            cout<<"Nhap ma: "; fflush(stdin); gets(macb);
            cout<<"Nhap ma don vi: "; fflush(stdin); gets(madv);
            cout<<"Nhap nam sinh: "; cin>>ns;
```

```
            cout<<"Nhap he so luong: "; cin>>hsl;
            cout<<"Nhap tien PC an ca (trieu dong): "; cin>>pc_anca;
            cout<<"Nhap bao hiem: "; fflush(stdin); gets(bh);
            }
            void xuat()
            {
                  cout<<"\nMa: "<<macb;
                  cout<<"\nMa don vi: "<<madv;
                  cout<<"\nNam sinh: "<<ns;
                  cout<<"\nHe so luong: "<<hsl;
                  cout<<"\nTien phu cap an ca (trieu dong): "<<pc_anca;
                  cout<<"\nBao hiem: "<<bh;
                  }
            float luong_cb()
            {
                  float luong_cb;
                  luong_cb=hsl*luongcb_cb+pc_anca;
                  return luong_cb;
                  }
      };

class giangvien: public canbo
{
      public:
            bool choice;
            float pc_lop;
            float pc_dochai;
      public:
```

```
void nhap()
{
        canbo::nhap();
        cout<<"Co phu cap doc hai khong? (1:Co - 0: Khong): ";
        cin>>choice;
        cout<<endl;
        }
void xuat()
{
        canbo::xuat();
        pc_lop=0.25*luongcb_gv;
        cout<<"\nTien phu cap lop (trieu dong): "<<pc_lop;
        if (choice==1)
        {
        pc_dochai=luongcb_gv*0.1;
        cout<<"\nTien PC doc hai (trieu dong): "<<pc_dochai;
        cout<<endl;
        }
}
float luong_gv()
{
        pc_lop=0.25*luongcb_gv;
        float luong_gv;
        luong_gv=(luongcb_gv*hsl)+pc_anca+pc_dochai+pc_lop;
        return luong_gv;
        }
};
```

```
void main()
{
      canbo a[50];
      giangvien b[50];
      int m,n,i;
      cout<<"Nhap so can bo: "; cin>>m;
      cout<<"Nhap so giang vien: "; cin>>n;
      cout<<"Nhap luong co ban can bo (trieu dong): "; cin>>luongcb_cb;
      cout<<"Nhap luong co ban gv (trieu dong): "; cin>>luongcb_gv;
      cout<<"\n---------------------\n";
      for (i=0; i<m; i++)
      {
            cout<<"Nhap thong tin can bo thu "<<i+1<<" \n";
            a[i].nhap();
            cout<<endl;
            }
      for (i=0; i<n; i++)
      {
            cout<<"Nhap thong tin giang vien thu "<<i+1<<" \n";
            b[i].nhap();
            cout<<endl;
            }
      cout<<"\nThong tin can bo va giang vien: ";
      for (i=0; i<m; i++)
      {
      a[i].xuat();
      cout<<"\nLuong can bo "<<i+1<<" (trieu dong): "<<a[i].luong_cb();
      cout<<endl;
```

```
        }
        for (i=0; i<n; i++)
        {
        b[i].xuat();
        cout<<"\nLuong GV "<<i+1<<" (trieu dong): "<<b[i].luong_gv();
        cout<<endl;
        }
        cout<<"\n\nDanh sach CB va GV phai dong thue thu nhap \n ";
        for (i=0; i<m; i++)
        {
              if (a[i].luong_cb()>5)
              {
                    a[i].xuat();
                    cout<<endl;
                    }
              }
        cout<<endl;
        for (i=0; i<n; i++)
        {
              if (b[i].luong_gv()>5)
              {
                    b[i].xuat();
                    cout<<endl;
                    }
              }
getch();
}
```

Câu 4.6: Lập chương trình thực hiện các việc sau:

- Tạo một lớp nhân viên với các dữ liệu gồm: Tên nhân viên, Đơn vị, Hệ số lương, Lương tối thiểu với các phương thức sau:
  - Nhập dữ liệu
  - Tính lương theo công thức: Lương chính = Lương tối thiểu * Hệ số lương.
- Hiện tại có nhu cầu tính lương mới theo cách sau:
  - Kỹ sư: Lương mới = Lương chính + Số năm trong nghề * Phụ cấp chuyên môn
  - Các nhân viên khác vẫn tính lương theo cách tính như trước

Hãy thiết kế chương trình thực hiện việc tính lương mới cho các nhân viên bằng phương pháp kế thừa chương trình đã có trước

```
#include <iostream.h>
#include <stdio.h>
#include <conio.h>

class nhanvien
{
 public:
      char tennv[25];
      char donvi[20];
      float hsl;
      float luongtt;
 public:
      void nhap()
      {
            cout<<"Nhap ten: "; fflush(stdin); gets(tennv);
            cout<<"Nhap don vi: "; fflush(stdin); gets(donvi);
            cout<<"Nhap he so luong: "; cin>>hsl;
```

```
            cout<<"Nhap luong toi thieu: "; cin>>luongtt;
            }
        float luong()
        {
            float luong;
            luong=luongtt*hsl;
            return luong;
            }
        void xuat()
        {
            cout<<"Ten: "<<tennv;
            cout<<"\nDon vi: "<<donvi;
            cout<<"\nHe so luong: "<<hsl;
            cout<<"\nLuong toi thieu: "<<luongtt;
            cout<<"\nLuong chinh: "<<luong();
            }
};

class kysu: public nhanvien
{
 public:
        int sonam;
        float pc_chuyenmon;
        int choice;
 public:
        void nhap()
        {
        nhanvien::nhap();
```

```
        cout<<"Co la ky su khong? (1:Co - 0: Khong) "; cin>>choice;
        if (choice==1)
            {
            cout<<"Nhap so nam trong nghe: "; cin>>sonam;
            cout<<"Nhap phu cap chuyen mon: "; cin>>pc_chuyenmon;
            }
            cout<<endl;
        }
        float luong_ks()
        {
            float luong_ks;
            luong_ks=luong()+sonam*pc_chuyenmon;
            return luong_ks;
            }
        void xuat()
        {
            nhanvien::xuat();
            if (choice==1)
            {
            cout<<"\nSo nam trong nghe: "<<sonam;
            cout<<"\nPhu cap chuyen mon: "<<pc_chuyenmon;
            cout<<"\nLuong ky su: "<<luong_ks();
            }
            cout<<endl;
        }
};
```

```
void main()
{
 kysu a[50];
 int n,i;
 cout<<"Nhap so nhan vien n= "; cin>>n;
 for (i=0; i<n; i++)
 {
       a[i].nhap();
       }
 for (i=0; i<n; i++)
 {
       a[i].xuat();
       cout<<endl;
       }
 getch();
}
```

<u>Câu 4.7:</u> Một nhà xuất bản nhận xuất bản sách. Sách có hai loại: loại có hình ảnh ở trang bìa và loại không có hình ảnh ở trang bìa. Loại có hình ảnh ở trang bìa thì phải thuê họa sĩ vẽ bìa.

- Viết chương trình thực hiện các yêu cầu :

  - Tạo một lớp cơ sở có tên là SACH để lưu thông tin về tên sách, tác giả, số trang, giá bán và định nghĩa hàm thành phần cho phép nhập dữ liệu, in dữ liệu cho các đối tượng của lớp SACH.
  - Tạo lớp BIA kế thừa từ lớp SACH để lưu các thông tin: Mã hình ảnh, tiền vẽ và định nghĩa hàm thành phần cho phép nhập, in dữ liệu cho các đối tượng của lớp BIA.

- Viết hàm main() thực hiện việc nhập xuất dữ liệu cho bài toán trên

```
#include <iostream.h>
#include <stdio.h>
#include <conio.h>

class SACH
{
 public:
      char tensach[25];
      char tacgia[20];
      int sotrang;
      float giaban;
 public:
      void nhap()
      {
            cout<<"Nhap ten sach: "; fflush(stdin); gets(tensach);
            cout<<"Nhap tac gia: "; fflush(stdin); gets(tacgia);
            cout<<"Nhap so trang: "; cin>>sotrang;
            cout<<"Nhap gia ban: "; cin>>giaban;
            }
      void in()
      {
            cout<<"Ten sach: "<<tensach;
            cout<<"\nTac gia: "<<tacgia;
            cout<<"\nSo trang: "<<sotrang;
            cout<<"\nGia ban: "<<giaban;
            }
 };
```

```
class BIA: public SACH
{
 public:
        int mahinh;
        float tienve;
        int choice;
 public:
        void nhap()
        {
                SACH::nhap();
                cout<<"Co phai la sach ve bia? (1:Co - 0:Khong) ";
                cin>>choice;
                if (choice==1)
                {
                cout<<"Nhap ma hinh anh: "; cin>>mahinh;
                cout<<"Nhap tien ve: "; cin>>tienve;
                }
                cout<<endl;
        }
        void in()
        {
                SACH::in();
                if (choice==1)
                {
                        cout<<"\nMa anh bia: "<<mahinh;
                        cout<<"\nTien ve: "<<tienve;
                        }
                cout<<endl;
```

```
            }
 };

void main()
{
 BIA a[50];
 int n,i;
 cout<<"Nhap so cuon sach n= "; cin>>n;
 for (i=0; i<n; i++)
 {
       cout<<"Nhap thong tin sach thu "<<i+1<<" \n";
       a[i].nhap();
       }
 for (i=0; i<n; i++)
 {
       cout<<"\nThong tin sach thu "<<i+1<<" \n";
       a[i].in();
       }
 getch();
}
```

<u>Câu 4.8:</u> Một nhà xuất bản nhận xuất bản sách. Sách có hai loại: loại có hình ảnh ở trang bìa và loại không có hình ảnh ở trang bìa. Loại có hình ảnh ở trang bìa thì phải thuê họa sĩ vẽ bìa.

- Viết chương trình thực hiện các yêu cầu :

- Tạo một lớp cơ sở có tên là SACH để lưu thông tin về tên sách, tác giả, số trang, giá bán và định nghĩa hàm thành phần cho phép nhập dữ liệu, in dữ liệu cho các đối tượng của lớp SACH.

- Tạo lớp BIA kế thừa từ lớp SACH để lưu các thông tin: Mã hình ảnh, tiền vẽ và định nghĩa hàm thành phần cho phép nhập, in dữ liệu cho các đối tượng của lớp BIA.
- Tạo lớp HOASY để lưu các thông tin họ tên, địa chỉ của họa sỹ và định nghĩa hàm thành phần cho phép nhập, in dữ liệu cho các đối tượng của lớp HOASY.
- Tạo lớp SACHVEBIA kế thừa từ lớp BIA và lớp HOASY và định nghĩa hàm thành phần cho phép nhập, in dữ liệu cho các đối tượng của lớp SACHVEBIA.

- Viết hàm main() cho phép nhập vào hai danh sách: danh sách các sách có vẽ bìa và danh sách các sách không có vẽ bìa (có thể dùng mảng tĩnh hoặc mảng con trỏ) để thực hiện, xuất dữ liệu ra màn hình.

```
#include <iostream.h>
#include <conio.h>
#include <stdio.h>

class sach
{
	public:
		char tensach[30];
		char tentg[25];
		int sotrang;
		float giaban;
	public:
		void nhap()
		{
		cout<<"Nhap ten sach: "; fflush(stdin); gets(tensach);
		cout<<"Nhap ten tac gia: "; fflush(stdin); gets(tentg);
		cout<<"Nhap so trang: "; cin>>sotrang;
		cout<<"Nhap gia ban: "; cin>>giaban;
		}
```

```
            void xuat()
            {
                  cout<<"\nTen sach: "<<tensach;
                  cout<<"\nTen tac gia: "<<tentg;
                  cout<<"\nSo trang: "<<sotrang;
                  cout<<"\nGia ban: "<<giaban;
                  }
      };

class bia: public sach
{
      public:
            int mahinh;
            float tienve;
      public:
            void nhap()
            {
                  sach::nhap();
                  cout<<"Nhap ma hinh anh: "; cin>>mahinh;
                  cout<<"Nhap tien ve: "; cin>>tienve;
            }
            void xuat()
            {
                  sach::xuat();
                  cout<<"\nMa hinh anh: "<<mahinh;
                  cout<<"\nTien ve hinh: "<<tienve;
                  cout<<endl;
                  }
```

```
	};

class hoasi
{
	public:
		char hoten[25];
		char diachi[30];
	public:
		void nhap()
		{
		cout<<"Nhap ho ten hoa si: "; fflush(stdin); gets(hoten);
		cout<<"Nhap dia chi hoa si: "; fflush(stdin); gets(diachi);
		}
		void xuat()
		{
		cout<<"Ten hoa si: "<<hoten;
		cout<<"\nDia chi hoa si: "<<diachi;
		}
	};

class sachvebia: public bia, public hoasi
{
	public:
		void nhap()
		{
			bia::nhap();
			hoasi::nhap();
			cout<<endl;
```

```
                }
            void xuat()
            {
                bia::xuat();
                hoasi::xuat();
                cout<<endl;
            }
        };

void main()
{
        int i, m, n;
        sachvebia a[50];
        sach b[50];
        cout<<"-Nhap so SACH VE BIA: "; cin>>m;
        cout<<"\n-Nhap so SACH KHONG VE BIA: "; cin>>n;
        cout<<"\n-----------------------\n";
        for (i=0; i<m; i++)
            {
                cout<<"Nhap thong tin SACH VE BIA thu "<<i+1<<" \n";
                a[i].nhap();
                }
        for (i=0; i<n; i++)
        {
            cout<<"Nhap thong tin SACH KHONG VE BIA thu "<<i+1<<" \n";
            b[i].nhap();
            }
        cout<<"\n-------------------\n";
```

```
    cout<<"-THONG TIN SACH VE BIA VA KHONG VE BIA ";
    cout<<"\n-Thong tin SACH VE BIA: \n";
    for (i=0; i<m; i++)
    {
        a[i].xuat();
        cout<<endl;
        }
    cout<<"\n-Thong tin sach KHONG VE BIA: \n";
    for (i=0; i<n; i++)
    {
        b[i].xuat();
        cout<<endl;
        }
    getch();
}
```

<u>Câu 4.9:</u> Lập chương trình thực hiện các việc sau:

- Tạo một lớp nhân viên với các dữ liệu gồm: Tên nhân viên, Đơn vị, Hệ số lương, Lương tối thiểu với các phương thức sau:
  - Nhập dữ liệu
  - Tính lương theo công thức: Lương chính = Lương tối thiểu * Hệ số lương.
- Hiện tại có nhu cầu tính lương mới theo cách sau:
  - Nhân viên: Lương mới = Lương chính + Số ngày làm trong tháng * Phụ cấp trách nhiệm
  - Các nhân viên khác vẫn tính lương theo cách tính như trước

Hãy thiết kế chương trình thực hiện việc tính lương mới cho các nhân viên bằng phương pháp kế thừa chương trình đã có trước

```
#include <iostream.h>
#include <stdio.h>
#include <conio.h>

class nhanvien
{
 public:
        char tennv[25];
        char donvi[20];
        float hsl;
        float luongtt;
 public:
        void nhap()
        {
              cout<<"Nhap ten: "; fflush(stdin); gets(tennv);
              cout<<"Nhap don vi: "; fflush(stdin); gets(donvi);
              cout<<"Nhap he so luong: "; cin>>hsl;
              cout<<"Nhap luong toi thieu: "; cin>>luongtt;
              }
        float luongchinh()
        {
              float lc;
              lc=luongtt*hsl;
              return lc;
              }
        void xuat()
        {
              cout<<"\nTen: "<<tennv;
```

```
            cout<<"\nDon vi: "<<donvi;
            cout<<"\nLuong chinh: "<<luongchinh();
            }
 };
class nhanvienmoi: public nhanvien
{
 public:
      int songay;
      float pc_trachnhiem;
      int choice;
 public:
 void nhap()
 {
      nhanvien::nhap();
      cout<<"Tinh luong theo cach moi hay cu (1:Moi, 0:Cu) ";
      cin>>choice;
      if (choice==1)
      {
      cout<<"Nhap so ngay lam trong thang: "; cin>>songay;
      cout<<"Nhap tien phu cap trach nhiem: "; cin>>pc_trachnhiem;
      }
      cout<<endl;
 }
 float luongmoi()
      {
            float luongmoi;
            luongmoi=luongchinh()+pc_trachnhiem*songay;
            return luongmoi;
```

```
            }
        void xuat()
        {
            nhanvien::xuat();
            if (choice==1)
            {
                cout<<"\nLuong moi: "<<luongmoi();
            }
            cout<<endl;
        }
};
void main() {
 nhanvienmoi a[50];
 int n,i;
 cout<<"Nhap so nhan vien n= "; cin>>n;
 for (i=0; i<n; i++)
 {
        cout<<"Nhap thong tin nhan vien thu "<<i+1<<" \n";
        a[i].nhap();
        }
 for (i=0; i<n; i++)
 {
        cout<<"\nThong tin nhan vien thu "<<i+1<<" \n";
        a[i].xuat();
        }
 getch();
}
```

CHƯƠNG 5: KHUÔN HÌNH

5.1 Tóm tắt lý thuyết

- Khuôn hình hàm

  Cú pháp:

```
template <danh sách tham số kiểu> <kiểu trả về>
tên hàm(khai báo tham số)
        {
          // định nghĩa hàm
    }
```

trong đó `<danh sách tham số kiểu>` là các kiểu dữ liệu được khai báo với từ khoá *class*, cách nhau bởi dấu phẩy. Kiểu dữ liệu là một kiểu bất kỳ, kể cả kiểu *class*.

- Khuôn hình lớp

Cú pháp:

```
template <class kieuso>  class SO
      {
          kieuso giatri;
            public :
                SO (kieuso x =0);
                void Hienthi();
          ...
      };
```

Cũng giống như các khuôn hình hàm, template `<class kieuso>` xác định rằng đó là một khuôn hình trong đó có một tham số kiểu kieuso. C++ sử dụng từ khoá **class** chỉ để nói rằng kieuso đại diện cho một kiểu dữ liệu nào đó.

5.2 Các dạng bài tập

Câu 5.1: Viết chương trình sử dụng khuôn hình hàm tìm giá trị nhỏ nhất của một mảng bất kỳ được nhập từ bàn phím

Hướng dẫn:

```
#include<iostream.h>
template <class T>
T nn(T a[], int n)
{ T min=a[1];
for(int i=1; i<=n;i++)
      if(a[i]<min)
      min=a[i];
return min;
}
main() {
int c[100];
float b[100];
int m,k,i;
cout<<"Nhap so phan tu nguyen: "; cin>>k;
for( i=1; i<=k;i++)
      cin>>c[i];
cout<<"Gia tri nho nhat cua mang nguyen la: "<<nn(c,k);
cout<<"\nNhap so phan tu thuc:"; cin>>m;
for( i=1; i<=m;i++)
      cin>>b[i];
cout<<"Gia tri nho nhat cua mang thuc la: "<<nn(b,m);
}
```

5.3 Các vấn đề về thảo luận, thực hành, thí nghiệm

Chương này cung cấp kiến thức chuyên sâu về lập trình hướng đối tượng là cách thức sử dụng khuôn hình để giải quyết các bài toán phức tạp. Kết thúc chương yêu cầu sinh viên sử dụng thành thạo cách sử dụng khuôn hình để giải các bài tập phức tạp, và luyện tập thông qua việc viết chương trình thực hiện các bài tập tổng hợp sau:

5.4 Bài tập sinh viên tự làm

Câu 5.2: Viết khuôn hình hàm tìm số lớn nhất của một mảng bất kỳ

```
#include <iostream.h>
#include <conio.h>

template <class x>
void tim_max(x a[], int n)
{
      x max;
      max=a[0];
      for (int i=1; i<n; i++)
      {
            if (max<a[i])
            max=a[i];
            }
            cout<<"Gia tri lon nhat trong mang: "<<max;
      }

void main()
{
      float a[50], n;
      cout<<"Nhap n= "; cin>>n;
```

```
        for (int i=0; i<n; i++)
        {
                cout<<"Nhap a["<<i<<"]= ";
                cin>>a[i];
                }
        tim_max(a,n);
        getch();
}
```

Câu 5.3: Sử dụng hàm template để in các giá trị nhỏ nhất của một mảng có kiểu số nguyên.

```
#include <iostream.h>
#include <conio.h>

template <class x>
void tim_min(x a[], int n)
{
        x min;
        min=a[0];
        for (int i=1; i<n; i++)
        {
                if (min>a[i])
                min=a[i];
                }
                cout<<"Gia tri nho nhat trong mang: "<<min;
        }
```

```
void main()
{
      int a[50], n;
      cout<<"Nhap n= "; cin>>n;
      for (int i=0; i<n; i++)
      {
            cout<<"Nhap a["<<i<<"]= ";
            cin>>a[i];
            }
      tim_min(a,n);
      getch();
}
```

<u>Câu 5.4:</u> Viết khuôn hình hàm để trả về giá trị trung bình của một mảng, các tham số hình thức của hàm này là tên mảng, kích thước mảng.

```
#include <iostream.h>
#include <conio.h>

template <class x>
x tbc(x a[], int n)
{
      float s=0;
      float tbc;
      for (int i=0; i<n; i++)
      {
            s=s+a[i];
            tbc=s/n;
            }
```

```
        return tbc;
        }

void main()
{
        float a[100], n;
        cout<<"Nhap so phan tu cua mang n= "; cin>>n;
        for (int i=0; i<n; i++)
        {
        cin>>a[i];
        }
        cout<<"TBC cua mang la: "<<tbc(a,n);
        getch();
}
```

- Viết một khuôn hình hàm (function template) tên maximum( ) mà trả về trị lớn nhất của hai đối số được truyền đến hàm khi hàm được gọi.
- Đưa khuôn hình hàm ở a. vào trong một chương trình C++ mà gọi hàm hai lần: một lần với 2 đối số kiểu số nguyên, một lần với 2 đối số thực chính xác đơn.

```
#include <iostream.h>
#include <conio.h>

template <class x>
x maximum(x a, x b)
{
        x max;
        if (a>b) max=a;
        else max=b;
```

```
	return max;
}

void main()
{
	int a,b;
	float c,d;
	cout<<"Nhap hai so nguyen a va b: "; cin>>a>>b;
	cout<<"So lon hon: "<<maximum(a,b);
	cout<<"\nNhap hai so thuc c va d: "; cin>>c>>d;
	cout<<"Son lon hon: "<<maximum(c,d);
	getch();
}
```

<u>Câu 5.5:</u> Viết khuôn hình hàm để sắp xếp kiểu dữ liệu bất kỳ theo chiều tăng dần.

```
#include <iostream.h>
#include <conio.h>

template <class x>
void sx(x a[], int n)
{
	x *p, *q;
	for (int i=0; i<n-1; i++)
	for (int j=i+1; j<n; j++)
	{
		p=&a[i];
		q=&a[j];
```

```
            if (*p>*q)
            {
                  x tg=*p;
                  *p=*q;
                  *q=tg;
                  }
            }
      }

void main()
{
      int a[50], n;
      cout<<"Nhap so phan tu n= "; cin>>n;
      for (int i=0; i<n; i++)
      {
            cin>>a[i];
            }
      sx(a,n);
      cout<<"Day sau khi sap xep: ";
      for (int i=0; i<n; i++)
      {
            cout<<a[i]<<" ";
            }
      getch();
}
```

Câu 5.6: Lập chương trình sử dụng khuôn hình hàm để tìm giá trị nhỏ nhất của dãy số bất kỳ.

```
#include <iostream.h>
#include <conio.h>
template <class x>
void tim_min(x value, int n)
{
      x min;
      for (int i=1; i<=n; i++)
      {
            cout<<"Nhap gia tri thu "<<i<<": ";
            cin>>value;
            if (i==1)
            {
                  min=value;
                  }
            else
            {
                  if (min>value)
                  min=value;
                  }
            }
      cout<<"\nMinimum value: "<<min;
}
```

```
void main()
{
	int n;
	cout<<"Nhap n= "; cin>>n;
	int value;
	tim_min(value,n);
	float value2;
	cout<<endl;
	tim_min(value2,n);
	getch();
}
```

<u>Câu 5.7:</u> Lập chương trình sử dụng khuôn hình hàm để tìm giá trị lớn nhất của dãy bất kỳ.

```
#include <iostream.h>
#include <conio.h>
template <class x>
void tim_max(x value, int n)
{
	x max;
	for (int i=1; i<=n; i++)
	{
		cout<<"Nhap gia tri thu "<<i<<": ";
		cin>>value;
		if (i==1)
		{
			max=value;
			}
```

```
            else
            {
                  if (max<value)
                  max=value;
                  }
            }
      cout<<"\nMaximum value: "<<max<<endl;
}

void main()
{
      int n;
      cout<<"Nhap n= "; cin>>n;
      int value;
      tim_max(value,n);
      float value2;
      tim_max(value2,n);
      getch();
}
```

<u>Câu 5.8:</u> Lập chương trình sử dụng khuôn hình hàm để sắp xếp kiểu dữ liệu bất kỳ theo chiều giảm dần.

```
#include <iostream.h>
#include <conio.h>
template <class x>
void sx(x a[], int n)
{
```

```
	for (int i=0; i<n-1; i++)
		for (int j=i+1; j<n; j++)
		{
			if (a[i]<a[j])
			{
				x tg=a[i];
				a[i]=a[j];
				a[j]=tg;
				}
			}
	}

template <class x>
void in(x a[], int n)
{
	for (int i=0; i<n; i++)
	cout<<a[i]<<" ";
	}

void main()
{
	int n;
	int a[50];
	cout<<"Nhap n= "; cin>>n;
	for (int i=0; i<n; i++)
	{
		cout<<"Nhap a["<<i+1<<"]= ";
		cin>>a[i];
```

```
            }
      sx(a,n);
      in(a,n);
      getch();
}
```

<u>Câu 5.9:</u> Lập chương trình xây dựng khuôn hình cho hàm tìm giá trị nhỏ nhất của hai số.

```
#include <iostream.h>
#include <conio.h>

template <class x1, class x2>
void min(x1 a, x2 b)
{
cout<<"So nho nhat la: ";
cout<<(a<b?a:b);
}

void main()
{
      int a, float b;
      cout<<"Nhap a= "; cin>>a;
      cout<<"Nhap b= "; cin>>b;
      min(a,b);
      getch();
}
```

Câu 5.10: Lập chương trình sử dụng khuôn hình hàm để tìm số lớn nhất của hai số bất kỳ.

```
#include <iostream.h>
#include <conio.h>
void max(x1 a, x2 b)
{
cout<<"So lon nhat la: ";
cout<<(a>b?a:b);
}

void main()
{
      int a, float b;
      cout<<"Nhap a= "; cin>>a;
      cout<<"Nhap b= "; cin>>b;
      max(a,b);
      getch();
}
```

Câu 5.11: Lập chương trình xây dựng khuôn hình lớp cho lớp điểm trong không gian 2 chiều. Yêu cầu xây dựng constructor để tạo đối tượng và phương thức xuất điểm ra màn hình.

```
#include <iostream.h>
#include <conio.h>

template <class x>
class diem
{
```

```
        private:
                x a,b;

        public:
                diem(x a1=0, x b1=0)
                {
                a=a1; b=b1;
                }
                void in()
                {
                        cout<<"(x= "<<a<<",y= "<<b<<")";
                        cout<<endl;
                        }
        };

void main()
{
        diem<float> c(1.2,2.4);
        c.in();
        diem<int> d(3,5);
        d.in();
        getch();
}
```

Câu 5.12: Xây dựng khuôn hình lớp tam giác có thuộc tính là độ dài ba cạnh, các phương thức: Constructor tạo đối tượng, hàm xuất dữ liệu.

```
#include <iostream.h>
#include <conio.h>

template <class x>
class tamgiac
{
      private:
            x a,b,c;
      public:
      tamgiac(x a1=0, x b1=0, x c1=0)
            {
            a=a1; b=b1; c=c1;
            }
      void xuat()
      {
      cout<<"Do dai cac canh: a= "<<a<<", b= "<<b<<", c= "<<c<<" ";
      cout<<endl;
      }
};


void main() {
      tamgiac<int> c(3,4,5);
      c.xuat();
      tamgiac<float> d(3.2,4.5,5.1);
```

```
        d.xuat();
        getch();
}
```

<u>Câu 5.13:</u> Lập chương trình sử dụng khuôn hình hàm để nhập vào một dãy số, xuất dãy số và tính trung bình cộng các phần tử của dãy số đó.

```
#include <iostream.h>
#include <conio.h>

template <class x>
void tbc(x value, int n)
{
        float tbc;
        float sum=0;
        for (int i=1; i<=n; i++)
        {
                cout<<"Nhap gia tri thu "<<i<<": ";
                cin>>value;
                sum=sum+value;
                }
        cout<<"Trung binh cong: "<<sum/n;
        }
void main()
{
        int value, n;
        cout<<"Nhap n= "; cin>>n;
        tbc(value,n);
        getch();
}
```

TÀI LIỆU THAM KHẢO

Tài liệu tham khảo chính:

[1] Lê Đình Hưng, Nguyễn Thanh Thuỷ, Lập trình hướng đối tượng với C++, NXB Khoa học và Kỹ thuật, 2005.

[2] Nguyễn Thanh Thuỷ, Nguyễn Quang Huy, Bài tập lập trình hướng đối tượng với C++, NXB Khoa học và Kỹ thuật, 2004.

Các tài liệu tham khảo khác:

[1] James P. Cohoon and Jack W.Davidson, C++ Program Design An Introduction to Programming and Object-Oriented Design, 2nd edition, WCB McGraw-Hill, 1999.

[2] Nell Dale, Chip Weems and Mark Headington, Programming and Problem Solving with C++, John & Barlett Publisher, 1996.

TNUT
THAI NGUYEN UNIVERSITY OF TECHNOLOGY
Address: 3-2 Street, ThaiNguyen City; Website: http://www.tnut.edu.vn