ĐẠI HỌC THÁI NGUYÊN

TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHIỆP

BỘ MÔN TIN HỌC CÔNG NGHIỆP

TÀI LIỆU THỰC HÀNH

HỌC PHẦN: **LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG**

MÃ SỐ HỌC PHẦN: TEE414

SỐ TÍN CHỈ: **01 TC**

ThS. Trần Thị Ngọc Linh

TÀI LIỆU THỰC HÀNH

HỌC PHẦN: **LẬP TRÌNH HƯỚNG ĐỐI TƯỢNG**

MÃ SỐ HỌC PHẦN: TEE414

SỐ TÍN CHỈ: **01 TC**

TRƯỞNG BỘ MÔN	**TRƯỞNG KHOA ĐIỆN TỬ**
(Ký và ghi rõ họ tên)	*(Ký và ghi rõ họ tên)*
TS. Nguyễn Văn Huy	***TS. Đào Huy Du***

MỤC LỤC

STT	Nội dung bài thực hành	Tiết TH
1	**Lớp**	**1,2**
2	**Hàm bạn**	**3,4**
3	**Toán tử tải bội**	**5,6**
4	**Kế thừa**	**7,8**
5	**Kế thừa**	**9,10**
6	**Khuân hình**	**11,12**

Bài 1. LỚP

1.1. Mục tiêu

- Khắc sâu khái niệm về class và cách sử dụng:
- Cách định nghĩa, khai báo lớp
- Đinh nghĩa các hàm thành phần trong lớp
- Khai báo public
- Khai báo private
- Hiểu và nắm được mục đích sử dung, cách hoạt động của lớp

1.2. Yêu cầu thiết bị

- Yêu cầu máy tính phải được cài đặt phần mềm BORLAND C++ 4.5 hoặc các phiên bản tương đương

1.3. Nội dung thực hiện

- Định nghĩa về Class
- Ý nghĩa và cách sử dụng
- Thực hành: Cho một bài đã được lập trình sẵn yêu cầu đoán kết quả thực hiện bài đó trên máy tính từ đó giúp hiểu rõ hơn về lớp và cách sư dụng

1.4. Nội dung chi tiết

Bài 1. 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()<< "\n"; cout <<p->get_b()<< "\n"; p++; }
cout <<"\n";
}
```

KQ: 1 2 3 4 5 6 7 8

Bài 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";
```

}

KQ: 0 1 2 3

Bài 3. 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 cho vector***
- ***In tọa độ của vector***
- ***Cộng, trừ, 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(); }
```

Bài tập sinh viên tự làm

Bài 1. Viết chương trình 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.

Bài 2. Viết chương trình thực hiện các công việc sau:

- Xây dựng lớp cơ sở MATHANG gồm:

- Thuộc tính: Tên hàng, năm sản xuất, giá thành
- Phương thức: Nhập, xuất thông tin

- Xây dựng lớp HDBANHANG kế thừa từ lớp MATHANG 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 các yêu cầu sau:
  - Nhập danh sách N hoá đơn bán hàng
  - Sắp xếp danh sách các mặt hàng có tiền lãi giảm dần
  - Hiện ra màn hình danh sách gồm: số thứ tự, tên hàng, giá thành, số lượng bán, giá bán, thành tiền, thuế và tiền lãi.
  - Tính tổng tiền của các hoá đơn bán hàng

Bài 2. HÀM BẠN, TOÁN TỬ TẢI BỘI

2.1. Mục tiêu:

- Giúp sinh viên khắc sâu kiến thức về Class
- Cách sử dụng Hàm bạn
- Giúp sinh viên biết cách sử dụng toán tử tải bội operator
- Vận dụng việc sử dụng toán tử tải bội trong các bài toán về Class

2.2. Yêu cầu thiết bị:

- Yêu cầu máy tính phải được cài đặt phần mềm BORLAND C++ 4.5 hoặc các phiên bản tương đương

2.3. Nội dung thực hành:

- Định nghĩa về Hàm bạn, Ý nghĩa và cách sử dụng
- Định nghĩa về toán tử tải bội, Ý nghĩa và cách sử dụng
- Thực hành: Lập trình trên máy tính ít nhất một bài toán cụ thể về hàm bạn, toán tử tải bội, từ đó giúp hiểu rõ hơn về nội dung kiến thức trên

2.4. Nội dung chi tiết

Bài 1. Viết chương trình thực hiện các công việc sau:

Tạo một lớp Complex để thực hiện các phép toán 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 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.
- Sử dụng hàm bạn 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<stdio.h>
class sp{
    public:
    float t,a;
    sp(float t1=0,float a1=0)
    {
        t=t1;a=a1;
    }
    void xuat()
    {
        cout<<"("<<t<<", "<<a<<")"<<endl;
            }
    friend sp operator+(sp a,sp b)
    {
        sp c;
        c.t=a.t+b.t;
        c.a=a.a+b.a;
        return c;
    }
    friend sp operator-(sp a,sp b)
    {
```

```
        sp c;
        c.t=a.t-b.t;
        c.a=a.a-b.a;
        return c;
    }
    friend sp operator*(sp a,sp b)
    {
        sp c;
        c.t=(a.t*b.t+a.a*b.a)/(b.t*b.t+b.a+b.a);
        c.a=(a.t*b.a+a.a*b.t)/(b.t*b.t+b.a*b.a);
        return c;
    }
    friend sp operator/(sp a,sp b)
    {
        sp c;
        c.t=(a.t*b.t-a.a*b.a)/(b.t*b.t+b.a+b.a);
        c.a=(a.t*b.a-a.a*b.t)/(b.t*b.t+b.a*b.a);
        return c;    }

};
main()
{float t2,a2,t1,a1;
```

```
    cout<<"nhap phan thuc va phan ao so phuc a:\n";
    cin>>t2>>a2;
    cout<<"nhap phan thuc va phan ao so phuc b:\n";
    cin>>t1>>a1;
        sp a(t2,a2),b(t1,a1),c;
    cout<<"so phuc a la:\n";
    a.xuat();
    cout<<"so phuc b la:\n";
    b.xuat();
    c=a+b;
    cout<<"tong 2 so phuc:\n";
    c.xuat();
    c=a-b;
    cout<<"hieu 2 so phuc:\n";
    c.xuat();
    c=a*b;
    cout<<"tich 2 so phuc:\n";
    c.xuat();
    c=a/b;
    cout<<"thuong 2 so phuc:\n";
    c.xuat();
}
```

Bài 2. Viết chương trình thực hiện các công việc sau:

1. Xây dựng lớp DIACHI gôm:

- Thuộc tính: tỉnh, huyện để lưu địa chỉ của nhân viên
- Phương thức: Nhập, xuất dữ liệu

2. Xây dựng lớp DATE gôm:

- Thuộc tính: ngày, tháng năm lưu ngày tháng năm sinh của nhân viên
- Phương thức: Nhập, xuất dữ liệu

3. Xây dựng lớp person kế thừa 2 lớp trên

- Thuộc tinh: số điện thoại
- Phương thức: Nhập xuất

4. Chương trình chính thực hiện:

- Nhập danh sách n nhân viên
- Hiển thị các thông tin của danh sách nhân viên vừa nhập
- Sử dụng toán tử tải bội operator sắp xếp danh sách nhân viên theo thứ tự ngày sinh tăng dần

```
#include<conio.h>

#include<stdio.h>

#include<string.h>

#include<iostream.h>

class diachi

{private:

  char *tinh,*huyen;

 public:

  diachi(char *t=NULL,char *h=NULL)
```

```
        {
         tinh=new char[strlen(t)+1];
         strcpy(tinh,t);
         huyen=new char[strlen(h)+1];
         strcpy(huyen,h);
        }
  ~diachi()
        {
         delete tinh;
         delete huyen;
        }
  void display()
        {
         cout<<"\t Tinh: "<<tinh;
         cout<<"\t Huyen: "<<huyen;
        }
};
class mydate
{private:
  int ng,th,n;
 public:
  mydate(int d=0,int m=0,int y=0)
```

```
        {
         ng=d; th=m; n=y;
        }
    void display()
        {
         cout<<"\t Ngay sinh: "<<ng<<"-"<<th<<"-"<<n;
        }
    int operator>(mydate &d)
        {
         if(n>d.n) return 1;
         if((n==d.n)&&(th>d.th)) return 1;
         if((n==d.n)&&(th==d.th)&&(ng>d.ng)) return 1; return 0;
        }
};
class person:public diachi,mydate
{private:
  char *ten;
  int phone;
 public:
  person(char *name=NULL,int d=0,int m=0,int y=0,char *t=NULL,char *h=NULL,int dt=0):diachi(t,h),mydate(d,m,y)
        {
```

```
            ten=new char[strlen(name)+1];
            strcpy(ten,name);
            phone=dt;
           }
  ~person()
           {
            delete ten;
           }
  void display()
           {
            cout<<"\n\n Ten: "<<ten;
            mydate::display();
            diachi::display();
            cout<<"\t Dien thoai: "<<phone;
           }
};
void main()
{clrscr();
 int i,j; const n=2;
 char ten[10],t[10],h[10];
 int ng,th,y,dt;
 person *a[n],*tg;
```

```
for(i=0;i<n;i++)
 {
  cout<<"\n ---------------------------";
  cout<<"\n Nhap phan tu thu "<<i+1<<" ";
  cout<<"\n Nhap ten: ";gets(ten);
  cout<<"\n Nhap ngay sinh: ";cin>>ng>>th>>y;
  cout<<"\n Nhap tinh: ";gets(t);
  cout<<"\n Nhap huyen: ";gets(h);
  cout<<"\n Nhap so phone: ";cin>>dt;
  a[i]=new person(ten,ng,th,y,t,h,dt);
 }
cout<<"\n\n Mang vua nhap la: ";
for(i=0;i<n;i++)
a[i]->display();
for(i=0;i<n-1;i++)
for(j=i+1;j<n;j++)
 if((mydate &) *a[i]>(mydate &) *a[j])
         {
          tg=a[i];
          a[i]=a[j];
          a[j]=tg;            }
cout<<"\n\n Mang sau khi sap xep tang dan theo ngay sinh: ";
```

```
for(i=0;i<n;i++)

a[i]->display();

getch();

}
```

Bài tập sinh viên tự làm

Bài 1. Viết chương trình thực hiện các công việc sau:

- Tạo một lớp **Point** dữ liệu kiểu nguyên để biểu diễn toạ độ của một điểm.
  - o Xây dựng một constructor để tạo đối tượng, constructor sử dụng các tham số có giá trị ngầm định.
  - o Xây dựng hàm thành phần public display() để thực hiện in điểm ra màn hình.
  - o Xây dựng hàm di chuyển điểm đến một toạ độ mới.
  - o Tạo lớp ColorPoint kế thừa lớp Point với dữ liệu kiểu nguyên để chỉ màu của điểm.
  - o Xây dựng hàm thành phần public để thực hiện in điểm có màu ra màn hình
  - o 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.

Bài 2. 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.

Bài 3: KẾ THỪA

4.1. Mục tiêu

- Giúp sinh viên biết thế nào là kế thừa, cách sử dụng đơn kế thừa, đa kế thừa
- Vận dụng việc sử dụng kế thừa trong việc giải quyết các bài toán về Class

4.2. Yêu cầu thiết bị

- Yêu cầu máy tính phải được cài đặt phần mềm BORLAND C++ 4.5 hoặc các phiên bản tương đương

4.3. Nội dung thực hành

- Định nghĩa về đơn kế thừa, đa kế thừa
- Ý nghĩa và cách sử dụng
- Thực hành: Lập trình trên máy tính ít nhất một bài toán cụ thể về kế thừa từ đó giúp hiểu rõ hơn về nội dung kiến thức trên

4.4 Nội dung chi tiết

Bài 1. Viết chương trình thực hiện các công việc sau:

Xây dựng chương trình minh họa việc quản lý kết quả thi của một lớp học không quá 100 sinh viên. Chương trình gồm 4 lớp:

- Lớp cơ sở sinh viên (sinhvien) chỉ lưu họ tên và số báo danh, các phương thức nhập, xuất dữ liệu.
- Lớp điểm thi (diemthi) kế thừa lớp sinh viên và lưu kết quả môn thi 1 và môn thi 2, các phương thức nhập, xuất dữ liệu.
- Lớp ưu tiên (uutien) lưu điểm ưu tiên của sinh viên, phương thức nhập, xuất dữ liệu.
- Lớp kết quả (ketqua) lưu tổng số điểm đạt được của sinh viên, phương thức xuất dữ liệu, phương thức hoán vị 2 sinh viên, sắp xếp danh sách sinh viên theo tổng điểm giảm dần.

Xây dựng hàm main để kiểm tra.

```
#include<conio.h>
#include<iostream.h>
#include<stdio.h>
class sinhvien
{
  public:
  char ht[40];
  char SBD[20];
  void nhap()
  {
        cout<<"\nTen sinh vien  :";
        gets(ht);
        cout<<"So bao danh     :";
         gets(SBD);
  }
  void xuat()
  {
              cout<<"\nTen sinh vien   :"<<ht;
        cout<<"\nSo bao danh     :"<<SBD;
  }
```

```
};
class diemthi:public sinhvien
{
  public:
  float mon1,mon2;
  void nhap()
  {
        sinhvien::nhap();
        cout<<"Mon thi 1 :  ";cin>>mon1;
        cout<<"Mon thi 2 :  ";cin>>mon2;
  }
  void xuat()
  {
        sinhvien::xuat();
        cout<<"\nMon thi 1 :  "<<mon1;
        cout<<"\nMon thi 2 :  "<<mon2;
  }
};
class uutien
{
  public:
  float diemuu;
```

```
        void nhap()
        {
        cout<<"Diem uu tien  :  "; cin>>diemuu;
        }
        void xuat()
        {
        cout<<"\nDiem uu tien  :  "; cout<<diemuu;
        }
};
class ketqua:public diemthi,public uutien
{
  public:
  int x,t;
  void nhap()
  {
        diemthi::nhap();
        uutien::nhap();
  }
  void xuat()
  {
        diemthi::xuat();
        uutien::xuat();
```

```
  }
  float tongdiem()
  {
        return(mon1+mon2+diemuu);
  }
};
main()
{
  ketqua a[10];
  int i,j,n,t,x;
  cout<<"Nhap so sinh vien vao:  "; cin>>n;
  for(i=0;i<n;i++)
  {
        cout<<"\n\nThong tin sinh vien thu "<<i+1;
        a[i].nhap();
  }
  cout<<"\nThong tin sinh vien vua nhap:";
  for(i=0;i<n;i++)
  {
        cout<<"\n\nSinh vien thu "<<i+1;
        a[i].xuat();
  }
```

```
  for(i=0;i<n-1;i++)
        for(j=i+1;j<n;j++)
              if (a[i].tongdiem()<a[j].tongdiem())
              {
        ketqua tg=a[i];
                    a[i]=a[j];
                    a[j]=tg;
              }
  cout<<"\nDanh sach tong diem giam dan\n";
  for(i=0;i<n;i++)
  a[i].xuat();
  cout<<"\n";
}
```

Bài tập sinh viên tự làm

Bài 1. Viết 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

Bài 2. 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 cũ

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

Bài 5: KHUÂN HÌNH

5.1. Mục tiêu

- Giúp sinh viên biết thế nào là khuân hình hàm, khuân hình lớp
- Vận dụng việc sử dụng khuân hình trong việc giải quyết các bài toán cụ thể

5.2. Yêu cầu thiết bị

- Yêu cầu máy tính phải được cài đặt phần mềm BORLAND C++ 4.5 hoặc các phiên bản tương đương

5.3. Nội dung thực hành

- Định nghĩa về khuân hình hàm, khuân hình lớp
- Ý nghĩa và cách sử dụng
- Thực hành: Lập trình trên máy tính ít nhất một bài toán cụ thể về khuân hình từ đó giúp hiểu rõ hơn về nội dung kiến thức trên

5.4. Nội dung chi tiết

Bài 1. Viết chương trình bằng ngôn ngữ C++ thực hiện các công việc sau:

Lập chương trình sử dụng khuôn hình hàm để sắp xếp kiểu dữ liệu bất kỳ theo chiều tăng dần.

```
#include<iostream.h>

#include<stdio.h>

template<class T>

void nhap(T a[100], int n)

{

   for(int i = 1; i <= n; i++)

   {

      cout << "nhap phan tu thu " << i << ":";
```

```
        cin >> a[i];
    }
}
template<class T>
void xuat(T a[100],int n)
{
    for(int i = 1; i <= n; i++)
    cout<<a[i];
}
template<class T>
void sx(T a[100], int n)
{
    for(int i=1;i<=n-1;i++)
            for(int j=i+1;j<=n;j++)
            {
                    if(a[i]>a[j])
                    {
                    T      tg=a[i];
                            a[i]=a[j];
                            a[j]=tg;
                    }
            }
```

```
}
    main()
    {
    int a[100];
    float b[100];
    int n,m;
    cout<<"nhap so luong phan tu mang nguyen:";
    cin>>n;
    nhap(a,n);
    cout<<"mang da sap xep:";
    sx(a,n);
    xuat(a,n);
    cout<<"nhap so luong phan tu mang thuc:";
    cin>>m;
    nhap(b,m);
    cout<<"mang da sap xep:";
    sx(b,m);
    xuat(b,m);
    }
```

Bài 2. 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>
```

```
template <class t>
class tamgiac
{
    t x,y,z;
    public:
    tamgiac( t x1=0 ,t y1=0,t z1=0)
    {
        x=x1;
        y=y1; z=z1 ;
    }
  void xuat()
    {
    cout <<x<<y<<z;
    }
};
main()
    {
    tamgiac <int> a(2,1,5);
    a.xuat();
    tamgiac <float> b(2.3,1.2,7);
    b.xuat();
    }
```

Bài tập sinh viên tự làm

Bài 1. Viết khuôn hình hàm tìm số lớn nhất của một mảng bất kỳ

Bài 2. Sử dụng hàm template để in các giá trị nhỏ nhất của một mảng có kiểu số nguyên.

Bài 3. 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.

Bài 4. Lập chương trình sử dụng khuôn hình hàm để sắp xếp kiểu dữ liệu bất kỳ theo chiều giảm dần.