# BÀI TẬP THỰC HÀNH 1

**1. Mục tiêu:**

### *Về kiến thức:*

- Sinh viên có thể lập trình giải các bài toán đơn giản bằng C++, vận dụng được các thức cơ bản trong C++: cấu trúc một chương trình C++, khai báo biến, các kiểu dữ liệu nguyên thuỷ, các phép toán cơ bản, các cú pháp điều khiển, các cú pháp chuyển vị…

### *Về kĩ năng:*

- Sinh viên phải biết cài đặt Dev-C++ và sử dụng được Dev-C++ để chạy một chương trình C++.

### *Về thái độ:*

- Tự giác chuẩn bị các câu hỏi và bài tập.

- Thực hiện các bài tập trên máy tính.

**2. Yêu cầu**

* Sinh viên chuẩn bị trước các bài tập thực hành,
* Thực hành các bài tập trong chương.

**3. Nội dung thực hành**

***3.1 Bài thực hành cơ bản***

***Bài 1***

Viết chương trình in ra giá trị tăng giảm của biến **i** cho trước.

***Bài 2***

Viết các chương trình nhập vào một số nguyên n và in ra số nhị phân tương ứng với n.

***Bài 3***:

Nhập vào ngày, thàng, năng và cho biết đó là ngày thứ bao nhiêu trong năm (biết rằng một năm có 365 ngày hoặc 336 ngày với năm nhuận).

***Bài 4***

Viết chương trình tìm các số nguyên tố nhỏ hơn một số n cho trước

a. Chỉ sử dụng chu trình **for**

b. Chỉ sử dụng chu trình **while**

c. Chỉ sử dụng chu trình **do-while**

***Bài 5*:**

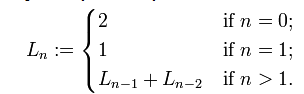
Xây dựng lớp **ToanHoc**, với n là số nguyên dương.

1. Kiểm tra xem một số nguyên n có phải là số nguyên tố hay không ?

2. Kiểm tra xem một nguyên n có phải là số hoàn hảo hay không ?

3. In ra các số chính phương <=n (n nhập tử bàn phím).

4. Tính phần tử thứ n của dãy Fibonaci: public: static int tinhFibo(int n)

Với công thức truy hồi được tính như sau:

***Bài 6***

1. Tính tổng dãy

S=1+2-3+……(-1)n+1.n

2. Tính tổng dãy:

S=1!+2!+3!+……+n!

1. Nhập n, nếu n lẻ : tính tổng các số lẻ <n, nếu n chẵn: tính tổng các số chẵn <n.
2. Tính tổng dãy



***3.2 Bài thực hành nâng cao***

***Bài 1:***

Tính tổng dãy:



# BÀI TẬP THỰC HÀNH 2

**1. Mục tiêu:**

### *Về kiến thức:*

- Sinh viên viết được các chương trình giải các bài toán theo cách tiếp cận hướng đối tượng với C++. Vận dụng được các kiến thức về lớp, đối tượng, phương thức C++.

### *Về kĩ năng:*

- Sinh viên phải biết cách tạo đối tượng, lớp,. biết tư duy bài toán theo tư tưởng hướng đối tượng.

### *Về thái độ:*

- Tự giác chuẩn bị các câu hỏi và bài tập.

- Thực hiện các bài tập trên máy tính.

1. **Yêu cầu**

* Sinh viên chuẩn bị trước các bài tập thực hành,
* Thực hành các bài tập trong chương.

1. **Nội dung thực hành**

***3.1 Bài thực hành cơ bản***

***Bài 1***

Viết chương trình:

1. Xây dựng lớp Nguoi gồm;

* Các thuộc tính chung : hoTen, diaChi, namSinh
* Các phương thức:
  + Phương thức: void nhap TT(){// nhập thông tin cho Nguoi}
  + Phương thức: void inTT(){//in thông tin cho Nguoi}

1. Xây dựng lớp VanDung chứa phương thức main(): Cho phép nhập thông tin của n người và hiển thị thông tin của n người vừa nhập ra màn hình.

***Bài 2:***

Thư viện của trường đại học KHTN có nhu cầu cần quản lý việc mượn sách. Sinh viên đăng ký và tham gia mượn sách thông qua các thẻ mượn mà thư viện đã thiết kế.

* Với mỗi thẻ mượn, có các thông tin sau: số phiếu mượn , ngày mượn, hạn trả, số hiệu sách, và các thông tin riêng về mỗi sinh viên đó.
* Các thông tin riêng về mỗi sinh viên đó bao gồm: Họ tên, năm sinh, tuổi, lớp.

1. Hãy xây dựng các lớp: SinhVien để quản lý các thông tin riêng về mỗi sinh viên, lớp TheMuon để quản lý việc mượn sách của mỗi đọc giả.

2. Viết chương trình nhập và hiển thị thông tin của một thẻ mượn.

***Bài 3:***

Để quản lý các biên lai thu tiền điện, người ta cần các thông tin như sau:

Với mỗi biên lai, có các thông tin sau: thông tin về hộ sử dụng điện, chỉ số cũ, chỉ số mới, số tiền phải trả của mỗi hộ sử dụng điện

Các thông tin riêng của mỗi hộ sử dụng điện gồm: Họ tên chủ hộ, số nhà, mã số công tơ của hộ dân sử dụng điện.

1. Hãy xây dựng các lớp: **KhachHang** để lưu trữ các thông tin riêng của mỗi hộ sử dụng điện, lớp **BienLai** để quản lý việc sử dụng và thanh toán tiền điện của các hộ dân. Xây dựng các phương thức nhập, và hiển thị thông tin của mỗi hộ sử dụng điện.

2. Cài đặt chương trình thực hiện các công việc sau:

+ Nhập vào các thông tin cho n hộ sử dụng điện

+ Tính tiền điện phải trả cho mỗi hộ dân, biết rằng tiền phải trả được tính theo công thức sau: *Số tiền phải trả=(Số mới - số cũ) \* 850000*.

***3.2 Bài thực hành nâng cao***

***Bài 1:***

1. Hãy xây dựng lớp DaGiac gồm có:

Các thuộc tính:

+ Số cạnh của đa giác

+ Mảng các số thực chứa kích thước các cạnh của đa giác

Các phương thức:

+ Nhập vào giá trị cho các cạnh của đa giác

+ Tính chu vi của đa giác

+ Hiển thị giá trị các cạnh của đa giác.

2. Xây dựng lớp TamGiac kế thừa từ lớp DaGiac, thực hiện ghi đè các phương thức cảu lớp TamGiac lên các phương thúc của lớp DaGiac.

3. Xây dựng hàm main() nhập kích thước của một tam giác và hiển thị chu vi, diện tích của tam giác đó.

# BÀI TẬP THỰC HÀNH 3

**1. Mục tiêu:**

### *Về kiến thức:*

- Sinh viên viết được các chương trình giải các bài toán theo cách tiếp cận hướng đối tượng với C++. Vận dụng được các kiến thức về lớp, đối tượng, phương thức trong C++.

### *Về kĩ năng:*

- Sinh viên phải biết cách tạo đối tượng, lớp,. biết tư duy bài toán theo tư tưởng hướng đối tượng.

### *Về thái độ:*

- Tự giác chuẩn bị các câu hỏi và bài tập.

- Thực hiện các bài tập trên máy tính.

1. **Yêu cầu**

* Sinh viên chuẩn bị trước các bài tập thực hành,
* Thực hành các bài tập trong chương.

1. **Nội dung thực hành**

***3.1 Bài thực hành cơ bản***

***Bài 1***

Xây dựng lớp ToanHoc có một biến thành phần là số nguyên n, với phương thức tĩnh (stattic) sau:

1. Kiểm tra xem một số nguyên k có phải là số nguyên tố hay không ?

bool kiemtraNguyenTo(int k)

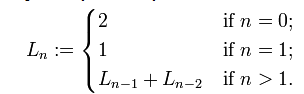
2. Kiểm tra xem một nguyên n có phải là số hoàn hảo hay không ?

bool kiemtraHoanHao(int n)

3. Kiểm tra xem một nguyên n có phải là số chính phương hay không ?

bool kiemtraSoChinhPhuong(int n)

4. Tính phần tử thứ n của dãy Fibonaci: public: static int tinhFibo(int n)

Với công thức truy hồi được tính như sau:

5. Tính tổng dãy



public: static double tinhTong(int n)

6. Tính tổng dãy



public: static double tinhTong(int n)

7. Tính tổng dãy

S=1+2-3+……(-1)n+1.n

public: static double tinhTong(int n)

8. Tính tổng dãy:

S=1!+2!+3!+……+n!

public: static double tinhTong(int n)

Xây dựng lớp VanDung có phương thức main() sử dụng các phương thức trong lớp ToanHoc ở trên.

***Bài 2:***

Viết chương trình xây dựng lớp Diem gồm:

1. Các thuộc tính double x,y tương ứng là hai giá trị hoành độ và tung độ và các phương thức:
2. Các phương thức tạo lập:
   1. *void Diem(){}*
   2. *void Diem(double x,double y){x=x1;y=y1}*
3. Các phương thức
4. *void inTTDiem(){ // in giá trị x,y ra màn hình }*
5. *double tinhKhoangCach(Diem d) {tính khoảng cách giữa điểm hiện thời và đối số d}*
6. Cài đặt lớp Main thực hiện tạo ra hai điểm *D1=new Diem(3,7); D2= new Diem(9,3)* và in khoảng cách của hai điểm trên.

***3.2 Bài thực hành nâng cao***

***Bài 1***

Viết chương trình: Xây dựng lớp SoPhuc gồm:

* 1. Các thuộc tính riêng gồm: phanThuc, phanAo kiểu double;
  2. Các phương thức:

+ Các toán tử tạo lập : SoPhuc(), SoPhuc(float pt, float pa)

+ Phương thức nhập vào một số phức: : void nhapSoPhuc()

+ Phương thức hiển thị một số phức: void inSoPhuc()

+ Phương thức cộng hai số phức : SoPhuc::congSoPhuc(SoPhuc sp)

+ Phương thức nhan hai số phức: SoPhuc::nhanSoPhuc(SoPhuc sp)

# BÀI TẬP THỰC HÀNH 4

**1. Mục tiêu:**

### *Về kiến thức:*

- Sinh viên viết được các chương trình giải các bài toán theo cách tiếp cận hướng đối tượng với C++. Vận dụng được các kiến thức về lớp, đối tượng, phương thức, truyền thông điệp trong C++, các khái niệm về toán tử tạo lập và lập trình sử dụng quan hệ kế thừa trong C++.

### *Về kĩ năng:*

- Sinh viên phải biết cách tạo đối tượng, lớp,. biết tư duy bài toán theo tư tưởng hướng đối tượng. Biết xây dựng các ví dụ sử dụng quan hệ kế thừa trong C++.

### *Về thái độ:*

- Tự giác chuẩn bị các câu hỏi và bài tập.

- Thực hiện các bài tập trên máy tính.

**2. Yêu cầu**

* Sinh viên chuẩn bị trước các bài tập thực hành,
* Thực hành các bài tập trong chương.

**3. Nội dung thực hành**

***3.1 Bài thực hành cơ bản***

***Bài 1:***

Viết chương trình:

1. Xây dựng lớp Nguoi gồm;

* Các thuộc tính chung : hoTen, diaChi, namSinh
* Các phương thức:
  + Các toán tử tạo lập : public: Nguoi()nội dung tạo lập }
  + Phương thức: public: void nhap TT(){// nhập thông tin cho Nguoi}
  + Phương thức: Public: void inTT(){//in thông tin cho Nguoi}

1. Xây dựng lớp NhanSu kế thừa từ lớp Nguoi và có thêm các thuộc tính: maNhanSu,heSochucVu, heSoLuong .
2. Các phương thức*: int luongCoBan và nhapTTNhanSu(),inTTNhanSu( ), void setLuongCoBan(int lcb){luongCoBan=lcb}, public: double tinhLuong(){ }*

theo tiêu chí:

lương=(heSoLuong+heSoChucVu)\*luongCoBan-heSoLuong\*luongCoBan\*5%,

Tại lớp Main tạo ra hai đối tượng thuộc lớp Nguoi, hai đối tượng thuộc lớp NhanSu nhập thông tin luongCoBan cho lớp NhanSu và nhập,in thông tin của 4 đối tượng trên

***Bài 2***

Xây dựng lớp Nguoi gồm có các thuộc tính chung : hoTen, diaChi, namSinh và các phương thức:

+Các toán tử tạo lập : public: Nguoi(){}, public: Nguoi(String ht,String dc,int ns){// nội dung tạo lập }

+Phương thức: public: void nhap TT(){// nhập thông tin cho Nguoi}

+Phương thức: Public: void inTT(){//in thông tin cho Nguoi}, public: void nhapTT(){// nhập thông tin từ bàn phím cho các thông tin của Nguoi }

Xây dựng lớp SinhVien kế thừa từ lớp người thêm các thuộc tính maSV, tenLop,diem1,diem2,diem3 và các phương thức:

+Các toán tử tạo lập.

+Tính điểm TB: public: float tinhDiemTrungBinh(){};

+viết đè hai phương thức nhapTT(), inTT()

Tại lớp lớp Main tạo ra một đối tượng thuộc lớp Nguoi, một đối tượng thuộc lớp SinhVien nhập thông tin diem1, diem2, diem3 cho lớp SinhVien và nhập,in thông tin của 2 đối tượng trên.

***3.2 Bài thực hành nâng cao***

***Bài 1:***

Để quản lý các hộ dân trong một khu phố, người ta quản lý các thông tin như sau:

* Với mỗi hộ dân, có các thuộc tính:

+ Số thành viên trong hộ ( số người)

+ Số nhà của hộ dân đó ( Số nhà được gắn cho mỗi hộ dân)

+ Thông tin về mỗi cá nhân trong hộ gia đình.

* Với mỗi cá nhân, người ta quản lý các thông tin như: họ và tên, tuổi, năm sinh, nghề nghiệp.

1. Hãy xây dựng các lớp: NhanSu để quản lý thông tin về mỗi cá nhân; lớp HoDan để quản lý thông tin về các hộ gia đình. Viết các phương thức để nhập, hiển thị thông tin cho mỗi cá nhân.

2. Xây dựng hàm main() cho phép nhập và hiển thị thông tin của 1 nhân sự.

# BÀI TẬP THỰC HÀNH 5

**1. Mục tiêu:**

### *Về kiến thức:*

- Sinh viên viết chương trình giải các bài toán về mảng một cách thành thạo. Khai thác được các thuộc tính và các phương thức của lớp mảng.

### *Về kĩ năng:*

- Sinh viên phải biết sử dụng mảng trong các bài tập.

### *Về thái độ:*

- Tự giác chuẩn bị các câu hỏi và bài tập.

- Thực hiện các bài tập trên máy tính.

**2. Yêu cầu**

* Sinh viên chuẩn bị trước các bài tập thực hành,
* Thực hành các bài tập trong chương.

**3. Nội dung thực hành**

***3.1 Bài thực hành cơ bản***

***Bài 1:***

Xây dựng lớp DaySo gồm:

* Các thuộc tính riêng (private) sau đây:
  + private: int n; //số phần tử của dãy
  + private: int m[] ; //lưu trữ các giá trị của dãy
* Các toán tử tạo lập: DaySo(int spt), DaySo(int m1[]), DaySo(){}
* Các phương thức:
  + public: void inDaySo() //in dãy ra màn hình
  + public: void nhapDaySo()//nhập dãy số từ bàn phím
  + public: void inSoNguyenTo(){} //in ra các số nguyên tố thuộc dãy số
  + public: void inSoHoanHao()
  + public: void sapXepTang() //sắp xếp dãy theo chiểu tăng
  + public: void sapXepGiam()//sắp xếp dãy theo chiều giảm
  + public: DaySo congDay(DaySo d1)// cộng hay dãy thành một dãy số

***Bài 2:***

Xây dựng lớp MaTran gồm:

* Các thuộc tính riêng(private)
  + private: int n,m;// số dòng và cột của ma trận
  + private: double M[][]; // lưu trữ các phần tử của ma trận
* Các phương thức tạo lập:
  + public: MaTran(){}
  + public: MaTran(int dong,int cot)
  + public: MaTran(double M1[][])
* Các phương thức:
  + public: void inMaTran()// in ma trận hiện thời
  + public: void nhapMaTran()// nhập ma trận từ bàn phím
  + public: bool kiemTraDoiXung()// kiểm tra tính đối xứng của mt qua đường chéo chính
  + public: congMaTran(MaTran M1)
  + public: nhanMaTran(MaTran M1)

***Bài 3:***

Viết chương trình định nghĩa lớp **Mang** với các thuộc được định nghĩa như sau:

+ private: int n; // số phần tử mảng

+ private: int A[]; // Danh sách các phần tử của mảng

1. Định nghĩa lớp **Mang** ở trên có các phương thức sau:

+ **nhapDL()** để nhập giá trị cho một mảng

+ **hienThiDL()** để hiện thị dữ liệu của mảng hiện thời

+ **lietKeNT** để hiển thị tất cả các số nguyên tố của mảng ra màn hình.

+ **lietKeHH** để hiển thị tất cả các số hoàn hảo của mảng ra màn hình.

+ **sapXep** để sắp xếp các phần tử của mảng theo thứ tự giảm dần

2. Cài đặt chương trình thực hiện nhập vào một mảng gồm n số nguyên ( n - nhập từ bàn phím thoả 2 ≤ n ≤ 100), sau đó hiển thị danh sách tất cả các số nguyên tố của mảng trên theo thứ tự tăng dần.

***3.2 Bài thực hành nâng cao***

***Bài 1:***

Thư viện của trường đại học KHTN có nhu cầu cần quản lý việc mượn sách. Sinh viên đăng ký và tham gia mượn sách thông qua các thẻ mượn mà thư viện đã thiết kế.

* Với mỗi thẻ mượn, có các thông tin sau: số phiếu mượn , ngày mượn, hạn trả, số hiệu sách, và các thông tin riêng về mỗi sinh viên đó.
* Các thông tin riêng về mỗi sinh viên đó bao gồm: Họ tên, năm sinh, tuổi, lớp.

1. Hãy xây dựng các lớp: SinhVien để quản lý các thông tin riêng về mỗi sinh viên, lớp TheMuon để quản lý việc mượn sách của mỗi đọc giả.

2. Viết chương trình nhập và hiển thị thông tin của một danh sách gồm m thẻ mượn, đếm số lượng thẻ mượn của lớp “K3A’

# BÀI TẬP THỰC HÀNH 6

**1. Mục tiêu:**

### *Về kiến thức:*

- Sinh viên viết được các chương trình giải các bài toán theo cách tiếp cận hướng đối tượng với C++. Vận dụng được các kiến thức về lớp, đối tượng, phương thức trong C++,

### *Về kĩ năng:*

- Sinh viên phải biết sử dụng và vận dụng các lớp đã có trong các bài tập.

### *Về thái độ:*

- Tự giác chuẩn bị các câu hỏi và bài tập.

- Thực hiện các bài tập trên máy tính.

**2. Yêu cầu**

* Sinh viên chuẩn bị trước các bài tập thực hành,
* Thực hành các bài tập trong chương.

**3. Nội dung thực hành**

***3.1 Bài thực hành cơ bản***

***Bài 1****:*

Viết chương trình:

1. *Xây dựng lớp SinhVien gồm:*

* Các thuộc tính:
  + String maSV, hoTen, diaChi;
  + double d1, d2, d3;
* Các phương thức:
  + Public: void nhapTTSV(), public: void inTTSV(),
  + public: double tinhDiemTrungBinh

1. *Xây dựng lớp DanhSachSinhVien gồm:*

* Các thuộc tính riêng:
  + int n;// số sinh viên
  + SinhVien dsSinhVien[];// lưu trữ thông tin của các sinh viên
* Các toán tử tạo lập: DanhSachSinhVien(int ssv)
  + - * DanhSachSinhVien(SinhVien mSinhVien[])
* Các phương thức:
  + Public: void nhapDanhSachSinhVien();
  + Public: void inDanhSachSinhVien();
  + Public: void inDanhSachSinhVien(String dc)//in danh sách sinh viên có địa chỉ là dc

***Bài 2****:*

Xây dựng lớp MaTran thực hiện thao tác sau:

1. Nhập ma trận.
2. Hiển thị ma trận.
3. Tìm hàng, cột hoặc đường chéo có tổng các phần tử lớn nhất.
4. Tìm ma trận chuyển vị của A
5. Tìm định thức của A
6. Tìm ma trận nghịch đảo của A

Xây dựng phương thức main() sử dụng các phương thúc trên.

***Bài 3:***

Viết chương trình quản lý khách hàng, mỗi khách hàng cần quản lý các thông tin : Họ tên, địa chỉ, điện thọai, mail. Số khách hàng tự nhập từ bàn phím, tìm khách hàng theo tên, địa chỉ.

***3.3 Bài thực hành nâng cao***

***Bài 1:***

Để quản lý các hộ dân trong một khu phố, người ta quản lý các thông tin như sau:

* Với mỗi hộ dân, có các thuộc tính:

+ Số thành viên trong hộ ( số người)

+ Số nhà của hộ dân đó ( Số nhà được gắn cho mỗi hộ dân)

+ Thông tin về mỗi cá nhân trong hộ gia đình.

* Với mỗi cá nhân, người ta quản lý các thông tin như: họ và tên, tuổi, năm sinh, nghề nghiệp.

1. Hãy xây dựng các lớp: NhanSu để quản lý thông tin về mỗi cá nhân; lớp HoDan để quản lý thông tin về các hộ gia đình. Viết các phương thức để nhập, hiển thị thông tin cho mỗi cá nhân.

2. Cài đặt chương trình thực hiện các công việc sau:

+ Nhập vào một dãy gồm n hộ dân (n - nhập từ bàn phím).

+ Hiển thị ra màn hình thông tin về các hộ trong khu phố.

# BÀI TẬP THỰC HÀNH 7

**1. Mục tiêu:**

### *Về kiến thức:*

- Sinh viên viết chương trình giải các bài toán về mảng và xâu một cách thành thạo.

### *Về kĩ năng:*

- Sinh viên phải biết sử dụng chuỗi trong các bài tập.

### *Về thái độ:*

- Tự giác chuẩn bị các câu hỏi và bài tập.

- Thực hiện các bài tập trên máy tính.

**2. Yêu cầu**

* Sinh viên chuẩn bị trước các bài tập thực hành,
* Thực hành các bài tập trong chương.

**3. Nội dung thực hành**

***3.1 Bài thực hành cơ bản***

***Bài 1:***

Viết chương trình

1.Xây dựng lớp Xau có thuộc tính dữ liệu là *pvivate String st* và các phương thức:

* Nhập dữ liệu : public: void nhapXau()
* In dữ liệu public: void inXau()
* Thay thế trong xâu hiện thời các xâu con st1 bằng xâu con st2
  + public: void thayThe(String st1,String st2)
* Sắp xếp một mảng xâu theo thứ tự giảm dần của từ điển như sau :
  + public: static sapXepXauGiam(String mangXau[])

1. Xây dựng lớp ChuongTrinh nhập vào n xâu từ bàn phím thực hiện sắp xếp rồi in kết quả ra màn hình

***Bài 2:***

Để xử lý các văn bản, người ta xây dựng lớp VanBan có thuộc tính riêng là một xâu ký tự. Hãy:

1. Xây dựng lớp VanBan có:

+ Các toán tử tạo lập : VanBan(), VanBan(String st).

+ Phương thức đếm số từ của xâu thuộc tính trong lớp hiện tại.

+ Phương thức đếm số từ kết thúc bởi ký tự ‘G’ (không phân biệt chữ hoa/thường).

2. Chuẩn hoá xâu thuộc tính trong lớp hiện tại theo các tiêu chí sau:

+ Ở đầu và cuối của xâu không có ký tự trống

+ Ở giữa xâu, tại vị trí bất kỳ không tồn tại hai ký tự trống đứng liền nhau.

***3.3 Bài thực hành nâng cao***

**Bài 1:**

1. Viết chương trình định nghĩa lớp **XauKyTu** với các phương thức thực hiện các công việc sau:

+ các toán tử tạo lập

+ Nhập dữ liệu cho xâu hiện thời

+ Đếm số từ của xâu

+ Chuẩn hoá xâu theo các tiêu chí sau: Ở đầu và cuối xâu không còn ký tự trống, Ở giữa xâu không tồn tại 2 ký tự trống đứng liền nhau).

+ Đếm số từ của xâu, số từ kết thúc bởi ký tự ‘G’ (không phân biệt hoa/thường)

+ Kiểm tra tính đối xứng của xâu

2. Cài đặt chương trình thực hiện nhập vào một xâu ký tự bất kỳ từ bàn phím, sau đó cho biết số từ của xâu vừa nhập; chuẩn hoá xâu rối hiển thị kết quả ra màn hình, cho biết số từ kết thúc bởi ký tự ‘G’ (không phân biệt hoa/thường) của xâu vừa nhập; In ra xâu đảo ngược của xâu hiện tại (nếu xâu không đối xứng).

# BÀI TẬP THỰC HÀNH 8

**1. Mục tiêu:**

### *Về kiến thức:*

- Sinh viên viết chương trình giải các bài toán thực tế sử dụng kiến thức tổng quan về C++.

### *Về kĩ năng:*

- Sinh viên phải biết sử dụng các thư viện C++ hỗ trợ trong các bài tập.

### *Về thái độ:*

- Tự giác chuẩn bị các câu hỏi và bài tập.

- Thực hiện các bài tập trên máy tính.

**2. Yêu cầu**

* Sinh viên chuẩn bị trước các bài tập thực hành,
* Thực hành các bài tập trong chương.

**3. Nội dung thực hành**

***3.1 Bài thực hành cơ bản***

***Bài 1:***

Xây dựng lớp PhanSo với hai thuộc tính riêng xác định tử số và mẫu số của phân số, đồng thời có các phương thức sau:

+ Các toán tử tạo lập

+ Các phép toán cộng, trừ, nhân, chia hai phân số

+ Phép kiểm tra một phân số có phải tối giản hay không

+ Phép tìm dạng tối giản của phân số

2. Viết một chương trình ứng dụng thực hiện việc nhập vào một dãy các phân số

+ In ra màn hình dạng tối giản của các phân số đó

+ Sắp xếp các phân số theo thứ tự tăng dần của tử số của mỗi phân số

# BÀI TẬP THỰC HÀNH 9

**1. Mục tiêu:**

### *Về kiến thức:*

- Sinh viên viết chương trình giải các bài toán thực tế sử dụng kiến thức tổng quan về C++.

### *Về kĩ năng:*

- Sinh viên phải biết sử dụng các thư viện C++ hỗ trợ trong các bài tập.

### *Về thái độ:*

- Tự giác chuẩn bị các câu hỏi và bài tập.

- Thực hiện các bài tập trên máy tính.

**2. Yêu cầu**

* Sinh viên chuẩn bị trước các bài tập thực hành,
* Thực hành các bài tập trong chương.

**3. Nội dung thực hành**

***Bài 1:***

Để quản lý các hộ dân trong một khu phố, người ta quản lý các thông tin như sau:

* Với mỗi hộ dân, có các thuộc tính:

+ Số thành viên trong hộ ( số người)

+ Số nhà của hộ dân đó ( Số nhà được gắn cho mỗi hộ dân)

+ Thông tin về mỗi cá nhân trong hộ gia đình.

* Với mỗi cá nhân, người ta quản lý các thông tin như: họ và tên, tuổi, năm sinh, nghề nghiệp.

1. Hãy xây dựng các lớp: NhanSu để quản lý thông tin về mỗi cá nhân; lớp HoDan để quản lý thông tin về các hộ gia đình. Viết các phương thức để nhập, hiển thị thông tin cho mỗi cá nhân.

2. Cài đặt chương trình thực hiện các công việc sau:

+ Nhập vào một dãy gồm n hộ dân (n - nhập từ bàn phím).

+ Hiển thị ra màn hình thông tin về các hộ trong khu phố.

# BÀI TẬP THỰC HÀNH 10

**1. Mục tiêu:**

### *Về kiến thức:*

- Sinh viên viết chương trình giải các bài toán thực tế sử dụng kiến thức tổng quan về C++.

### *Về kĩ năng:*

- Sinh viên phải biết sử dụng các thư viện C++ hỗ trợ trong các bài tập.

### *Về thái độ:*

- Tự giác chuẩn bị các câu hỏi và bài tập.

- Thực hiện các bài tập trên máy tính.

**2. Yêu cầu**

* Sinh viên chuẩn bị trước các bài tập thực hành,
* Thực hành các bài tập trong chương.

**3. Nội dung thực hành**

***3.1 Bài thực hành cơ bản***

***Bài 1:***

Một công ty được giao nhiệm vụ quản lý các phương tiện giao thông gồm các loại: ô tô, xe máy, xe tải.

+ Mỗi loại phương tiện giao thông cần quản lý: Số máy, hãng sản xuất, năm sản xuất, giá bán.

+ Các ô tô cần quản lý: số chỗ ngồi, kiểu động cơ

+ Xe máy cần quản lý: công suất

+ Xe tải cần quản lý: trọng tải.

1. Xây dựng các lớp XeTai, XeMay, OTo kế thừa từ lớp PTGT; đồng thời có các hàm để truy nhập, hiển thị và kiểm tra các thuộc tính của các lớp.

2. Xây dựng lớp QLPTGT cài đặt các phương thức thực hiện các chức năng sau:

+ Nhập thông tin đăng ký cho một danh sách gồm m phương tiện

+ Nhập vào số máy của một phương tiện, cho biết phương tiện đó thuộc loại gì? Và hiển thị thông tin quản lý của phương tiện đó.