آشنایی با ارث بری (inheritance)

یکی از مباحث بسیار مهم در شی گرایی ، وراثت میباشد. وراثت به ما اجازه میدهد تا یک کلاس را با ویژگی یک کلاس دیگر تعریف کنیم! به کد زیر دقت کنید :

```
class Triangle{
    public:
        void triangle(){
            cout<<"I am a triangle\n";
        }
};
```

کلاس Triangle (مثلث) ما یک تابع با نام triangle() دارد . حال ما یک کلاس به صورت مشتق (ارث برده شده) از کلاس Triangle (مثلث) میسازیم. اسم آن را Isosceles (متساوی الساقین) مینامیم :

```
class Isosceles : public Triangle{
    public:
        void isosceles(){
            cout<<"I am an isosceles triangle\n";
        }
};
```

حال ما با ساخت شی از کلاس مشتق شده و استفاده از آن میتوانیم به توابع کلاس اصلی (مثلث) دسترسی داشته باشیم :

```
int main(){
    Isosceles isc;
    isc.isosceles();
    isc.triangle();
    return 0;
}
```

خروجی کد بالا به فرم زیر است :

```
I am an isosceles triangle
I am a triangle
```

اکنون یک تابع درون کلاس Isosceles بنویسید که جواب خروجی ما به فرم زیر باشد :

```
I am an isosceles triangle
In an isosceles triangle two sides are equal
I am a triangle
```

شما باید کد زیر را تکمیل کنید تا پاسخ این سوال را دهید :

```
#include <cmath>
#include <cstdio>
#include <vector>
#include <iostream>
#include <algorithm>
using namespace std;

class Triangle {
public:
    void triangle() {
        cout << "I am a triangle\n";
    }
};
class Isosceles : public Triangle {
public:
    void isosceles() {
        cout << "I am an isosceles triangle\n";
    }
    //Write your code here.
};
int main() {
    Isosceles isc;
    isc.isosceles();
```

```
        isc.description();
        isc.triangle();
        return 0;
    }
```

چند مضربی (inheritance)

به شما سه کلاس A , B , C داده میشود . هر سه این کلاس ها تابع func مخصوص به خود را دارند.

در کلاس A : این تابع مقداری که به آن پاس داده میشود را دوبرابر میکند.

```
class A
{
    public:
        A(){
            callA = 0;
        }
    private:
        int callA;
        void inc(){
            callA++;
        }

    protected:
        void func(int & a)
        {
            a = a * 2;
            inc();
        }
    public:
        int getA(){
            return callA;
        }
};
```

در کلاس B : این تابع مقداری که به آن پاس داده میشود را سه برابر میکند.

```
class B
{
    public:
```

```
        B(){
            callB = 0;
        }
    private:
        int callB;
        void inc(){
            callB++;
        }
    protected:
        void func(int & a)
        {
            a = a * 3;
            inc();
        }
    public:
        int getB(){
            return callB;
        }
};
```

در کلاس C : این تابع مقداری که به آن پاس داده میشود را پنج برابر میکند.

```
class C
{
    public:
        C(){
            callC = 0;
        }
    private:
        int callC;
        void inc(){
            callC++;
        }
    protected:
        void func(int & a)
        {
            a = a * 5;
            inc();
```

```
        }
    public:
        int getC(){
            return callC;
        }
};
```

به شما کلاس D به فرم زیر داده میشود :

```
class D
{

    int val;
    public:
        //Initially val is 1
         D()
         {
             val = 1;
         }

         //Implement this function
         void update_val(int new_val)
         {

         }
         //For Checking Purpose
         void check(int); //Do not delete this line.
};
```

حال شما باید تابع update_val را جوری بنویسید که مقدار val موجود در کلاس D را به new_val تغییر دهد . این کار باید تنها با صدا زدن توابع $*func*$ موجود در کلاس های A , B , C انجام دهید. تضمین میشود که new_val تنها مضربی از 2 ، 3 ، 5 باشد.

ورودی :

حاوی تنها یک خط است که مقدار new_val را از کاربر میگیرد

خروجی :

به صورت اتوماتیک توسط کد زیر خروجی داده خواهد شد که فرمت نمونه آن در مثال ها موجود میباشد .

مثال :

ورودی :

```
30
```

خروجی :

```
Value = 30
A's func called 1 times
B's func called 1 times
C's func called 1 times
```

توضیحات :

در ابتدا مقدار val یک میباشد . سپس تابع func موجود در کلاس A اجرا میشود و و مقدار val دوبرابر میشود. سپس تابع func موجود در کلاس B اجرا میشود و مقدار val سه برابر میشود. سپس تابع func موجود در C اجرا میشود و این بار 5 برابر میشود که میشود 30 !

کد خروجی شما باید به فرم زیر باشد :

```
#include<iostream>

using namespace std;

class A
{
    public:
```

```
        A(){
            callA = 0;
        }
    private:
        int callA;
        void inc(){
            callA++;
        }

    protected:
        void func(int & a)
        {
            a = a * 2;
            inc();
        }
    public:
        int getA(){
            return callA;
        }
};

class B
{
    public:
        B(){
            callB = 0;
        }
    private:
        int callB;
        void inc(){
            callB++;
        }
    protected:
        void func(int & a)
        {
            a = a * 3;
            inc();
        }
    public:
        int getB(){
```

```
            return callB;
        }
};

class C
{
    public:
        C(){
            callC = 0;
        }
    private:
        int callC;
        void inc(){
            callC++;
        }
    protected:
        void func(int & a)
        {
            a = a * 5;
            inc();
        }
    public:
        int getC(){
            return callC;
        }
};

/*******************************/

class D
{

    int val;
    public:
        //Initially val is 1
         D()
         {
             val = 1;
         }
```

```
        //Implement this function
        void update_val(int new_val)
        {

        }
        //For Checking Purpose
        void check(int); //Do not delete this line.
};

/*****************************/

void D::check(int new_val)
{
    update_val(new_val);
    cout << "Value = " << val << endl << "A's func called " << getA() << " times " <<
}

int main()
{
    D d;
    int new_val;
    cin >> new_val;
    d.check(new_val);

}
```

کلاس تاریخ - زمان (inheritance, operator overloading)

پیش‌نیاز مسئله: زمان شیئ است ۲ (مباحث پیشرفته شیئ گرایی)

تو این سوال می خواهیم کلاس **تاریخ - زمان** درست کنیم که از 2 کلاس **زمان** و **تاریخ** ارث بری می کند. و از شما می خواهیم این کلاس ها را پیاده سازی کنید.

کلاس `Time`

این کلاس همان کلاسی است که در تمرین جلسه ششم سوال *زمان شئی است ۲* پیاده سازی کردید. از آن استفاده بکنید.

کلاس `Date`

این کلاس شامل سه متغییر می باشد که به شکل زیر است:

- سال
- ماه
- روز

متد های این کلاس عبارت است از:

- با استفاده از عملگر `>>` یک شی از ورودی بگیرد
- با استفاده از عملگر `<<` شی از کلاس `Date` را در خروجی چاپ کند
- با استفاده از عملگر های `=>` , `>` , `<` , `=<` بتواند 2 شی از این کلاس را مقایسه کند
- همچنین با استفاده از overloading شی از این کلاس را به رشته تبدیل کند
- با استفاده از عملگر `+` ۲ شی از این تابع را باهم جمع کند.

کلاس `DateTime` این کلاس از ۲ کلاس بالایی ارث بری می کند و شامل متغییر های زیر می باشد:

- سال
- ماه
- روز

- ساعت
- دقیقه

متد های این کلاس عبارت است از:

- با استفاده از عملگر >> یک شی از ورودی بگیرد
- با استفاده از عملگر << شی از کلاس DateTime را در خروجی چاپ کند
- با استفاده از عملگر های <= , < , > , >= بتواند 2 شی از این کلاس را مقایسه کند
- همچنین با استفاده از overloading شی از این کلاس را به رشته تبدیل کند
- با استفاده از عملگر + ۲ شی از این تابع را باهم جمع کند.

در طراحی کلاس DateTime به **موارد استثنا** از جمله موارد زیر دقت کنید:

- در صورتی که دقیقه و ساعت در بازه صحیح نبود، نزدیک ترین عدد را جایگزین کند.
- در صورت خطا در روز و یا ماه با استفاده از assertion ها خطا بدهد.

خانواده شیپولی (square-rectangle (liskov) problem)

برنامه ای برای اشکال مختلف بنویسید. اشکال زیر در این برنامه وجود دارند :

۱. مثلث
۲. مستطیل
۳. لوزی
۴. مربع
۵. دایره (ورودی مرکز و شعاع)

هر یک از این اشکال با مجموعه ای از نقاط بر روی مختصات دکارتی مشخص میشود. پس ما در هر کلاس مجموعه ای از نقاط(اعشاری) داریم! از طرفی دیگر هر شکل دارای یک رنگ میباشد که در هنگام ساخت (سازنده) هر شی رنگ آن را مشخص میکنیم. اسم هر رنگ نهایتا 16 کاراکتر است. حال با کمک این نقاط وظیفه ی ما محاسبه موارد زیر برای هر شکل میباشد :

۱. محیط
۲. مساحت
۳. مرکز شکل

در این برنامه ما ابتدا شکلی که قصد داریم بسازیم انتخاب میکنیم و سپس نقاط مربوط به شکل هارا از کاربر گرفته و تمام موارد بالا را در خروجی برای آن شکل نشان میدهیم.

ورودی

در خط اول تعداد شکل ها می آید. سپس در خط بعدی نوع تمام شکل هارا دریافت میکنیم . و بعد از آن برای هر شکل نقاط لازمه را دریافت میکنیم !

خروجی

برای هر شکل باید سه مورد ذکر شده تا *دو رقم اعشار* به فرم مثال ها نشان داده شود.

مثال

ورودی:

```
4
1 2 3 5
GREEN : 0:0 2:0 1:4
RED : 0:0 0:8 2:8 2:0
LIGHTBLUE : 0:2 1:0 2:2 1:4
BLUE : 0:0 10
```

خروجی:

```
GREEN : 10.25 4.00 1:1.33
RED : 20.00 16.00 1:4
LIGHTBLUE : 8.94 4.00 1:2
BLUE : 62.83 314.15 10
```

مورد تحقیقی

با توجه به اینکه میدونیم هر مربع یک مستطیل است، در اینجا رابطه بین مربع و مستطیل در طراحی این کلاس چیست؟ درباره رابطه ی درست بین مربع و مستطیل تحقیق کنید و در یک فایل PDF بنویسید و به همراه کد سوال آپلود کنید.

می توانید از این لینک هم استفاده کنید.

نکات :

۱. تمامیه متغییر ها باید عضو خصوصی باشند.
۲. کلاس های شما باید به بهترین نوع ممکن ارث برده باشند. (با توجه به تحقیق)
۳. مرکز مثلث همان مرکز ثقل مثلث خواهد بود که از میانگین مختصات نقاطش بدست می‌آید.