subclass

Subclass is a class derived from another.

```
class subclass : public superclass{
 //code
};
```

Example A - Rectangle

```
class Shape{
    private:
        int width;
        int height;
    public:
        void set(int w,int h){width=w;height=h;}
};
class Rectangle : public Shape{
    public:
        int getArea(){return width*height;}
};
```

Example A - Rectangle

```
#include <iostream>
using namespace std;
int main(){
 Rectangle rec;
 rec.set(5,6);
 cout<<getArea();
}
```

http://cpp.sh/4osl

Example A - Error!

Since width,length are private in Shape, subclass Rectangle cannot access them. How to solve it?

Example A - Error!

Since width,length are private in Shape, subclass Rectangle cannot access them. How to solve it? **protected!!**

```
class Shape{
    protected://Only for subclass
    int width;
    int height;
    public:
    set(int w,int h){width=w;height=h;}
};
class Rectangle : public Shape{
    public:
        int getArea(){return width*height;}
};
```

http://cpp.sh/6ehq

Practice-FloatFraction Class

Create a new class Floatclass class based on the Fraction class. Floatclass class has one extra public member function **double get_float()** which returns decimal number of the fraction. `http://cpp.sh/3jpv`

- Subclass cannot access private members of Superclass.
- Consider a type conversion carefully.

```
int value=1.23;
int x=static_cast<int>(value*10.0);
```

Check- FloatFraction Class

```
class Fraction{
    protected:
        int num,den;
        //code
}
class FloatFraction:public Fraction{
    public:
        double get_float(){
            return static_cast<double>(num)/den;
        }
};
```

http://cpp.sh/7tkx

Check2- FloatFraction Class

If you do not use "protected", you need to use public functions "getn(), getd()".

```
class Fraction{
    private:
        int num,den;
        //code
}
class FloatFraction:public Fraction{
    public:
        double get_float(){
            double x=getn();
            return x/getd();
        }
};
```

http://cpp.sh/9px3

Constructor of subclass

Theorem

Subclass do not inherit constructors from their Superclass.

Constructor of subclass

Theorem

Subclass do not inherit constructors from their Superclass.

Theorem (C++11)

C++11 allows to inherit Constructor from Superclass.

Constructor of subclass

Theorem

Subclass do not inherit constructors from their Superclass.

Theorem (C++11)

C++11 allows to inherit Constructor from Superclass.

```
class FloatFraction:public Fraction{
    public:
        using Fraction::Fraction;//Inherit Constructors.
        double get_float(){
            double x=getn();
            return x/getd();
        }
};
```

Inadequate Constructor Inheritance

```
class Point2D{
    protected:
        int x,y;
    public:
        Point2D(){x=0;y=0;}
        Point2D(int nx,int ny){x=nx;y=ny;}
};
class Point3D:public Point2D{
    protected:
        int z;
    public:
    using Point2D::Point2D;
    // How about z?
};
```

None of the inherited constructors set a value of z.

Check-Inadequate Constructor Inheritance

```
class Point2D{
    protected:
        int x,y;
    public:
        Point2D(){x=0;y=0;}
        Point2D(int nx,int ny){x=nx;y=ny;}
};
class Point3D:public Point2D{
    protected:
        int z=0;//c++11
    public:
    using Point2D::Point2D;
    Point3D(int nx,int ny,int nz):Point2D(nx,ny),z(nz){}//c++11
};
```

http://cpp.sh/3tcz

Private,Protected inheritance

```
class A {public: int x;
protected: int y;
private: int z;};
class B : public A
{   // x is public
    // y is protected
    // z is not accessible from B};
class C : protected A
{    // x is protected
    // y is protected
    // z is not accessible from C};
class D : private A
{    // x is private
    // y is private
    // z is not accessible from D};
```

http://stackoverflow.com/a/1372858