

```

#include <iostream>
using namespace std;

class B{
    public:
        void f() { cout<<"f() de B"<<endl; }
        virtual void g() { cout<<"g() de B"<<endl; }
        virtual void h() = 0;
    protected:
        int b;
};

class D1 : virtual public B{
    public:
        void f() { cout<<"f() de D1"<<endl; }
        virtual void g() { cout<<"g() de D1"<<endl; }
    protected:
        int d1;
};

class D2 : virtual public B{
    public:
        void f(int i) { cout<<"f("<<i<<") de D2"<<endl; }
        virtual void h() { cout<<"h() de D2"<<endl; }
    private:
        int d2;
};

class D3 : public D1{
    public:
        void g() { cout<<"g() de D3"<<endl; }
        void h() { cout<<"h() de D3"<<endl; }
    private:
        int d3;
};

class D4 : public D1, public D2{
    private:
        int d4;
};

void f(B& b){
    cout<< "f() externa" << endl;
    b.f();
    b.g();
    b.h();
}

int main(){
    B b, *pb;
    D1 d1; D2 d2; D3 d3; D4 d4;
    f(b); f(d1); f(d2); f(d3); f(d4);
    d4.D1::f();
    d4.f(5);
    d4.f(3.7);
    d4.g();
}

```

```

        d4.h();
        pb = new D4;
        pb->f();
        pb->D4::f(3);
        pb->f();
        pb->g();
        pb->h();

        delete pb;
    }

```

- Corrija los errores, si los hay, en la definición de las clases B, D1, D2, D3 y D4. Para cada clase enumere sus miembros declarados o definidos explícitamente.
- Diga si el programa provoca algún error de compilación o de ejecución y por que. Si lo tiene, modifique el código adecuadamente para solucionarlo.

## 2. Dada la clase paramétrica

```

template <class T1, class T2>
class par
{
public:
    par() : prime(T1()), segun(T2()) {}
    par(const T1& x, const T2& y) : prime(x), segun(y) {}
    T1 primero() const { return prime; }
    T2 segundo() const { return segun; }
    T1& primero() { return prime; }
    T2& segundo() { return segun; }

    //ostream& operator <<(ostream& os, par<T1, T2>& parxy);
private:
    T1 prime;
    T2 segun;
};

```

- Sobrecargue operator << para la plantilla de clase **par**
- Defina un tipo racional a partir de ella para representar números racionales con el signo en el numerador. Sobrecargue el operador de suma de números racionales.
- Defina una clase **complejo** con partes real e imaginaria de tipo **double**. Sobrecargue el operador de autosuma (+=) de números complejos.