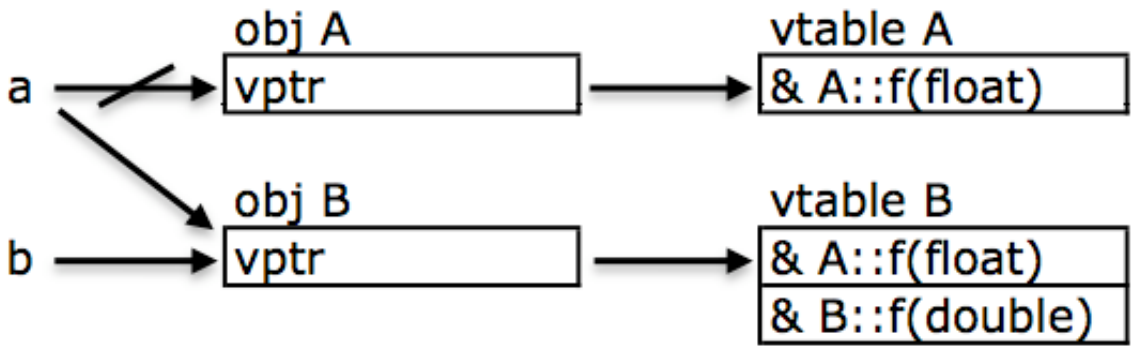
DISPENSA DI ESERCIZI C++ CON SOLUZIONI DI “LINGUAGGI DI PROGRAMMAZIONE”

Autore: Alberto De Bortoli
A.A. 2008-‘09
versione: 1.3
data: 9/12/2008

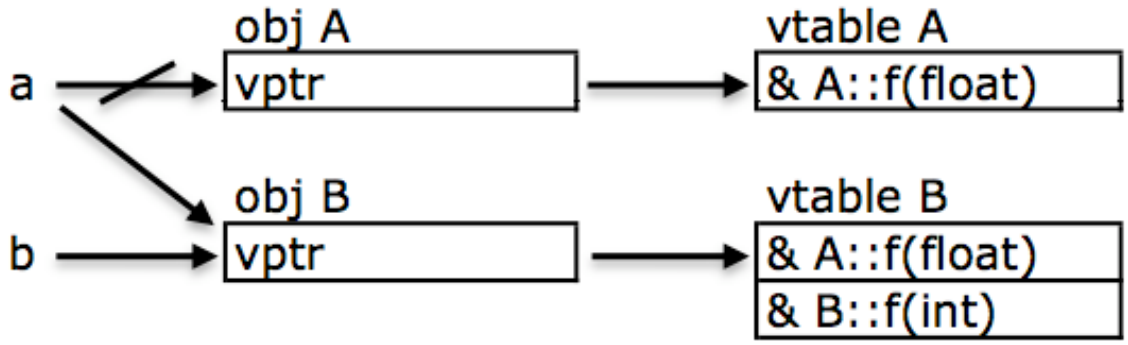
Esercizio 1a

```
#include<iostream>
using namespace std;
class A{
    public:
      virtual void f(float x){cout<<1<<endl;}};
class B: public A{
    public:
      virtual void f(double x){cout<<3<<endl;}};

main()
{
  A* a=new A;
  B* b=new B;
  a=b;
  b->f(3.14f);
  a->f(3.14);
}

/* stampa 3 1 */
```



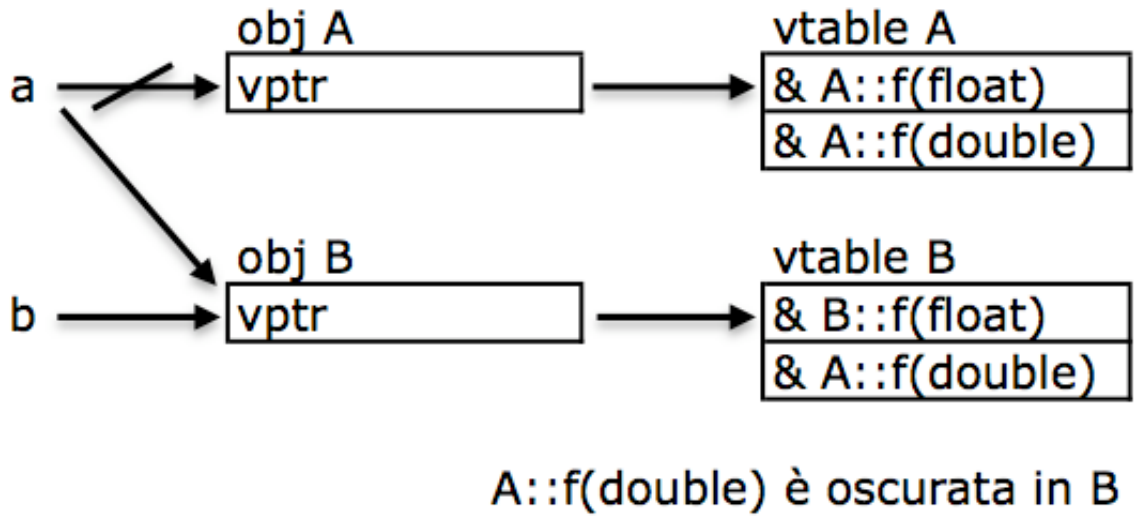
Esercizio 1b

```
#include<iostream>
using namespace std;
class A{
    public:
      virtual void f(float x){cout<<1<<endl;}};
class B: public A{
    public:
      virtual void f(int x){cout<<2<<endl;}};

main()
{
  A* a=new A;
  B* b=new B;
  a=b;
  a->f(3);
  b->f(3);
  a->f(3.14f);
  b->f(3.14f);

}

// warning: passing ‘float’ for argument 1 to ‘virtual void A::f(int)’

/* stampa 1 2 1 2 */
```



Esercizio 2a

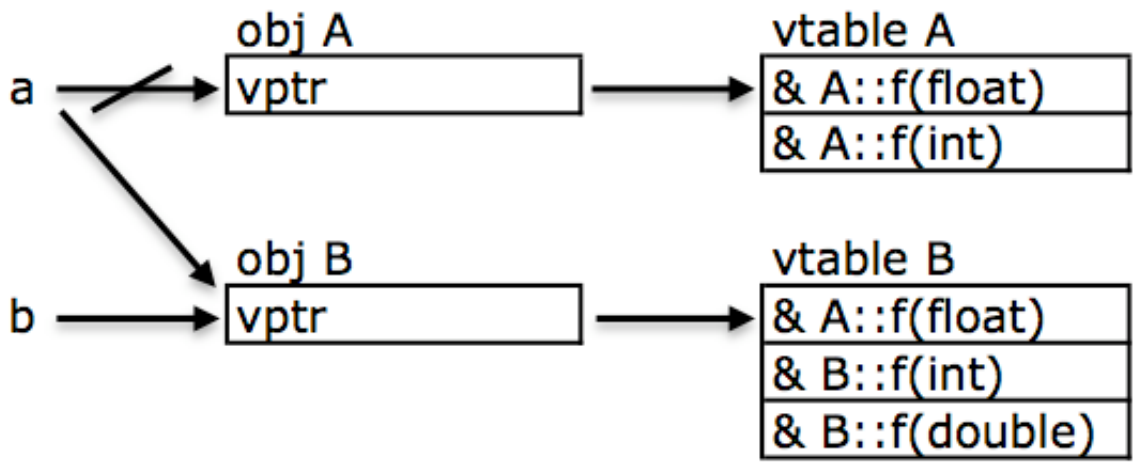
```
#include<iostream>
using namespace std;
class A{
    public:
      virtual void f(float x){cout<<1<<endl;}
      virtual void f(double x){cout<<2<<endl;}};
class B: public A{
    public:
      virtual void f(float x){cout<<3<<endl;}};

main()
{
  A* a=new A;
  B* b=new B;
  a->f(3.14f);
  a=b;
  b->f(3.14);
  a->f(3.14);
}


/* stampa 1 3 2 */
```



A::f(double) è oscurata in B

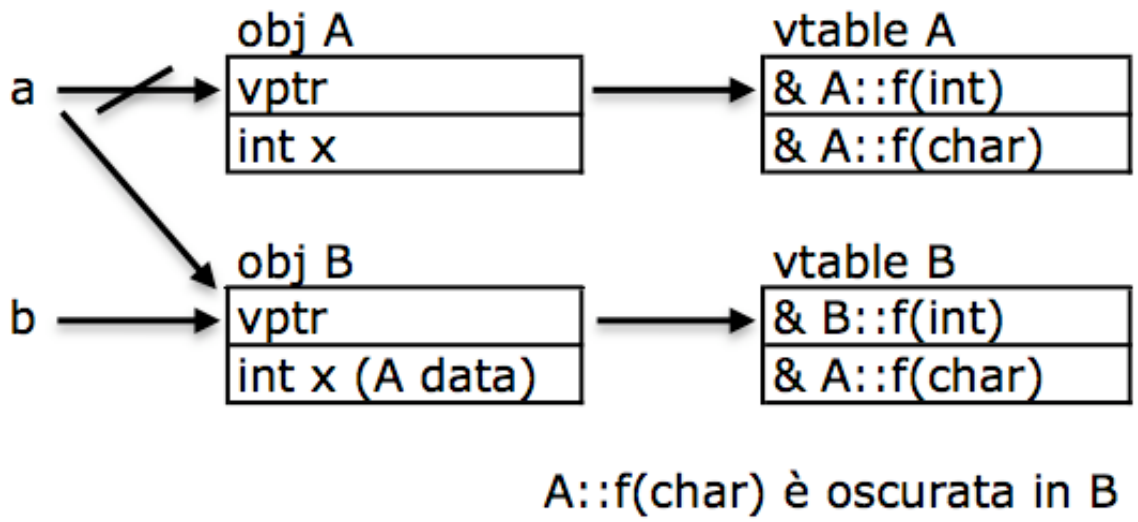
Esercizio 2b

```
#include<iostream>
using namespace std;
class A{
    public:
      virtual void f(float x){cout<<1<<endl;}
      virtual void f(int x){cout<<2<<endl;}};
class B: public A{
    public:
      virtual void f(double x){cout<<3<<endl;}
      virtual void f(int x){cout<<4<<endl;}};

main()
{
  A* a=new A;
  B* b=new B;
  a->f(3.14f);
  a=b;
  b->f(3.14);
  //a->f(3.14);
  //compile-time error: call of overloaded ‘f(double)’ is ambigous
  //note: candidates are: virtual void A::f(float)
  //note: virtual void A::f(int)
  a->f(3.14f);
  b->f(3);
  a->f(3);
}

/* stampa 1 3 1 4 4 */
```



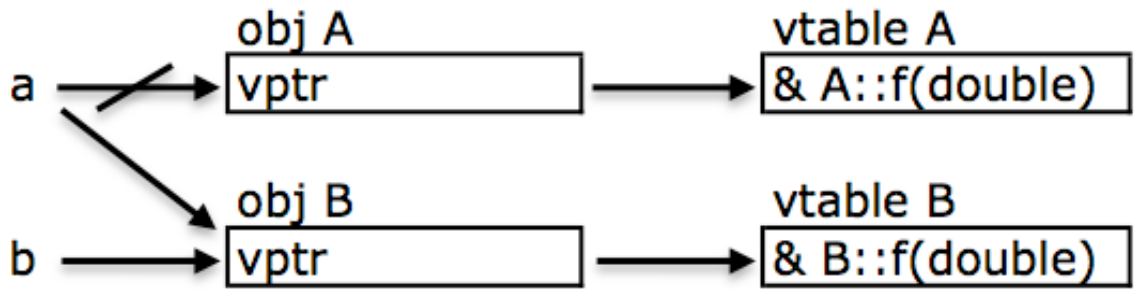
Esercizio 3

```
#include<iostream>
using namespace std;

class A{
    public:
      int x;
      A(int y){x=y;}
      virtual void f(int i){cout<<"A-int"<<endl;}
      virtual void f(char i){cout<<"A-char"<<endl;}};

class B: public A{
    public:
      B(int y):A(y){}
      virtual void f(int i){cout<<"B-int"<<endl;}};

main() {
    A* a=new A(2);
    B* b=new B(10);
    a->f('a');
    a=b;
    a->f('a');
    b->f('a');
}

/* stampa
A-char
A-char
B-int
*/
```



Esercizio 4

```
#include<iostream>
using namespace std;

class A{
    public:
      void f(int x){cout<<1<<endl;}
      virtual void f(double x){cout<<2<<endl;}};

class B: public A{
    public:
      virtual void f(double x){cout<<3<<endl;}};

main()
{
    A* a=new A;
    B* b=new B;
    a->f(5);
    a->f(5.0);
    a=b;
    a->f(5); //richiama la f non virtual, jump al codice della funzione
    b->f(5); //dovrebbe chiamare A::f(int) e quindi fare un jump
      //ma chiamandosi f (cioè ha lo stesso nome) è stata oscurata
    a->f(5);
    a->f(5.0);
    b->f(5.0);
    b->A::f(5.0);
    b->A::f(5);
}

// warning: passing ‘double’ for converting 1 to ‘virtual void
B::f(int)’

/* stampa 1 2 1 3 1 3 3 2 1 */
```



Esercizio 5

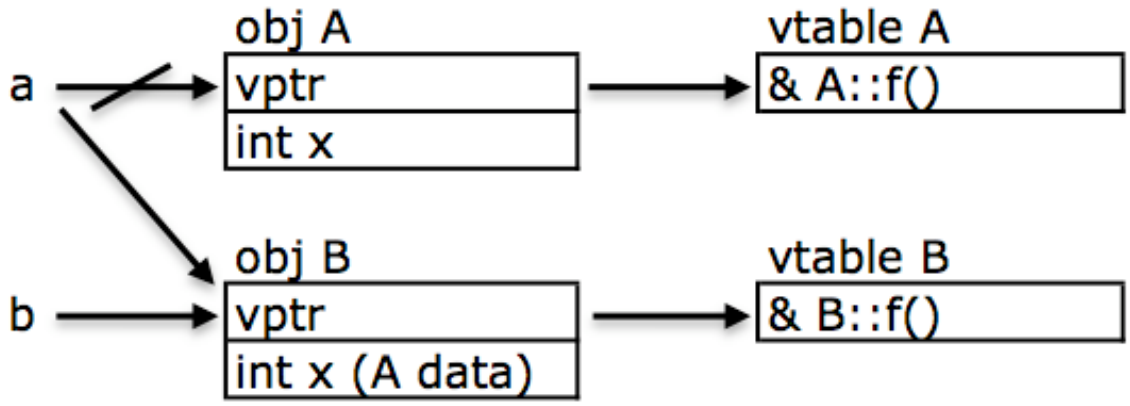
```
#include<iostream>
using namespace std;

class A{
    public:
      void f(){cout << 1 << endl;}
      void g(){f(); cout << 2 << endl;}};

class B: public A{
    public:
      void f(){cout <<3 << endl;}
      void g(){f(); cout << 4 << endl;}};

main(){
    A* a;
    B* b=new B();
    a=b;
    a->g();
    b->g();
    }

/* stampa 1 2 3 4 */
```

Esercizio 6

```
#include<iostream>
using namespace std;

class A{
    public:
      int x;
      A(int y){x=y;}
      void f(){cout<<1;}
      void g(){f(); cout<<2<<x<<endl;}};

class B: public A{
    public:
      B(int y):A(y){}
      void f(){cout<<3;}
      void g(){f(); cout<<4<<x<<endl;}};

main(){
    A* a=new A(5);
    B* b=new B(10);
    a=b;
    a->g();
    }

/* stampa 1 2 10 */
```

Esercizio 7

```
#include<iostream>
using namespace std;

class A{
    public:
      int x;
      A(int y){x=y;}
      virtual void f(){cout<<1;}
      void g(){f(); cout<<2<<x<<endl;}};

class B: public A{
    public:
      B(int y):A(y){}
      virtual void f(){cout<<3;}};

main(){
    A* a=new A(5);
    B* b=new B(10);
    a->g();
    a=b;
    a->g();
    }

/* stampa 1 2 5 e 3 2 10 */
```

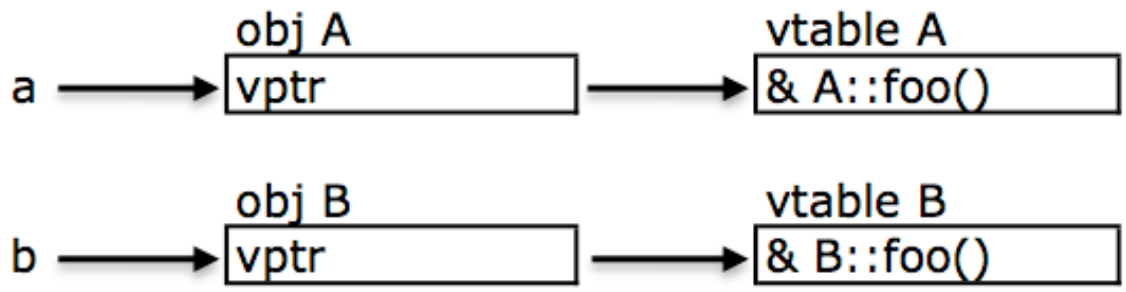


Esercizio 8

```
#include<iostream>
using namespace std;

class A{
    public:
    virtual void foo(){cout<<"A::foo\n";}
    };

class B:public A{
    public:
    virtual void foo(){cout<<"B::foo\n";}
    };

void test(A a)
    {a.foo();}

main()
{
  A a;
  B b;
  test(a);
  test(b);  // si fa cast da b ad a
}

/* stampa
   A::foo
   A::foo
 */
```

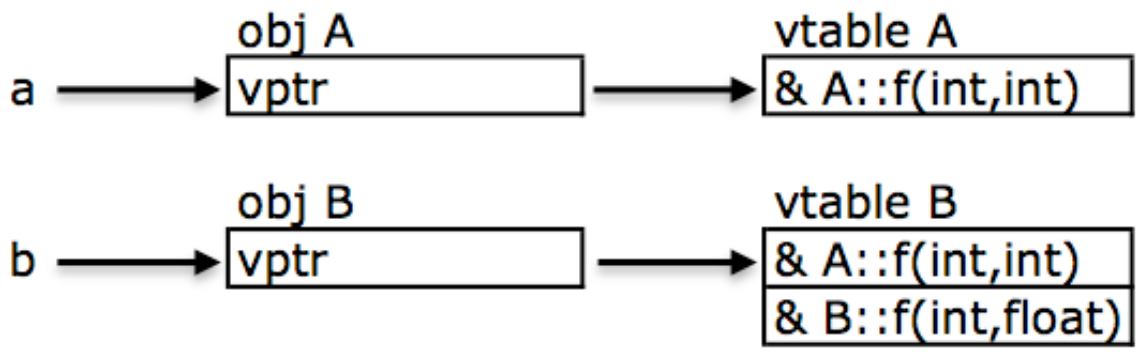
Esercizio 9

```
#include<iostream>
using namespace std;

class A{
    public:
      virtual void foo(){cout<<"A::foo\n";}
    };

class B:public A{
    public:
      virtual void foo(){cout<<"B::foo\n";}
    };

void test(A* a)
    {a->foo();}

main()
{
  A* a=new A;
  B* b=new B;
  test(a);
  test(b);
}

/* stampa
   A::foo
   B::foo
 */
```



Esercizio 10

```
#include<iostream>
using namespace std;

class A{
    public:
      virtual void f(int x, int y){cout<<x+y<<" A::f(int,int)\n";}
};

class B:public A{
    public:
      virtual void f(int x, float y){cout<<x+y<<" B::f(int, float)\n";}
};


void test(A* a)
    {a->f(3,3.5f);}

main()
{
    A* a=new A;
    B* b=new B;
    test(a);
    test(b);
}

/* stampa
   6 A::f(int,int)
   6 A::f(int,int)
 */
```



B::f(int, float) oscura A::f(int,int) in B

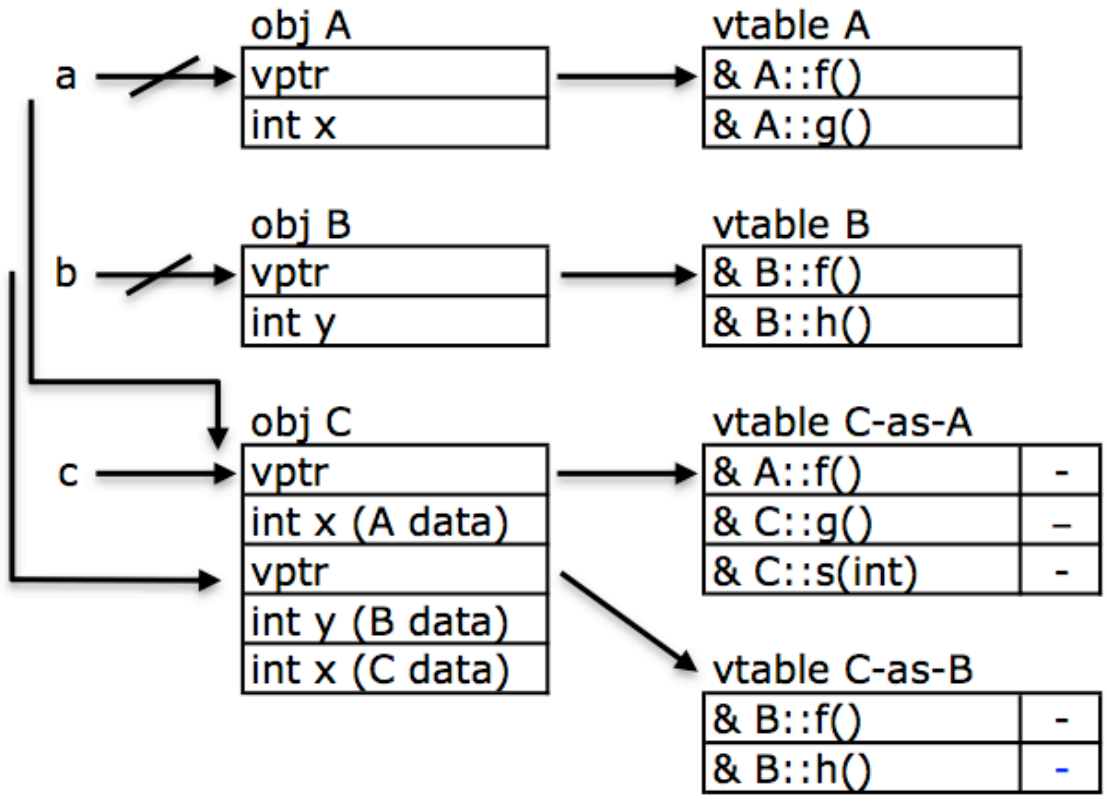
Esercizio 11

```
#include<iostream>
using namespace std;

class A{
    public:
      int x;
      A(int y){x=y;}
      void f(int i){cout<< x<<endl;}
      void g(char i){cout<< x<<endl;} };

class B: public A{
    public:
      B(int y):A(y){}
      void a(int i){cout<<"B-int"<<endl;} };

main() {
    A* a = new A(2);
    B* b = new B(10);
    a->g('a');
    b->g('a');
}


// stampa: 2 10
```

Esercizio 12

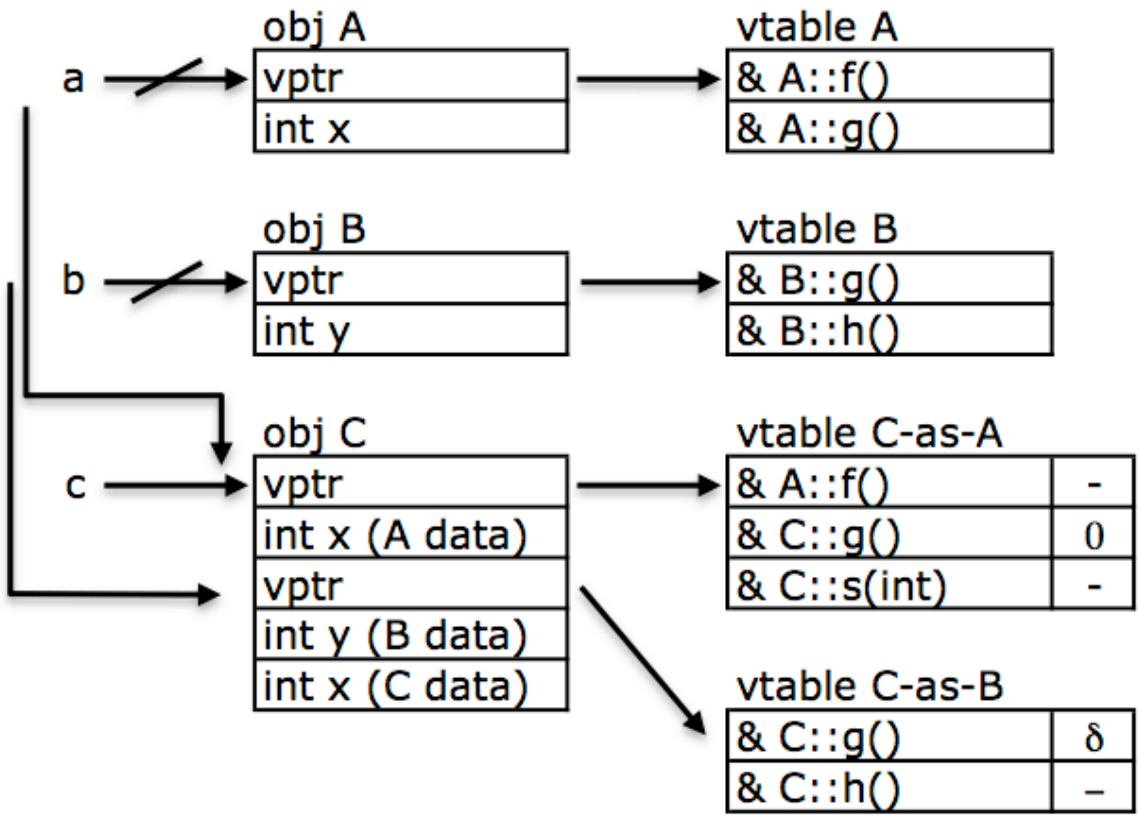
```
//esame 21 giugno 2007

#include<iostream>
using namespace std;

class A{
    public:
      int x;
      A(int a=0){x=a;}
      void r(int x) {f();}
      virtual void f() {cout<<1<<endl;}
      virtual void g() {cout<<2<<endl;}};

class B{
    public:
      int y;
      B(int b=0){y=b;}
      virtual void f(){cout<<3<<endl;}
      virtual void h(double a){cout<<"h di B"<<endl;}};

class C: public A, public B{
    public:
      int z;
      C(int a=0, int b=0, int c=0):A(a), B(b){z=c;}
      //virtual void f(){cout<<0<<endl;}
      virtual void g(){cout<<4<<endl;}
      virtual void s(int x){cout<<5<<endl;}};

main(){

A*a; B*b; C*c=new C(1,2,3);
a=c;
b=c;
a->f();
b->f();
//c->f();   //name clash
/*compile-time error: request for member ‘f’ is ambiguous
  candidates are: virtual void B::f()
                  virtual void A::f() */
c->r(3);

}
```

```
/*
 stampa 1 3 1
 se fosse presente anche la f() in C e non fosse commentata c->f(),
 stamperebbe 0 0 0 0
 */
```



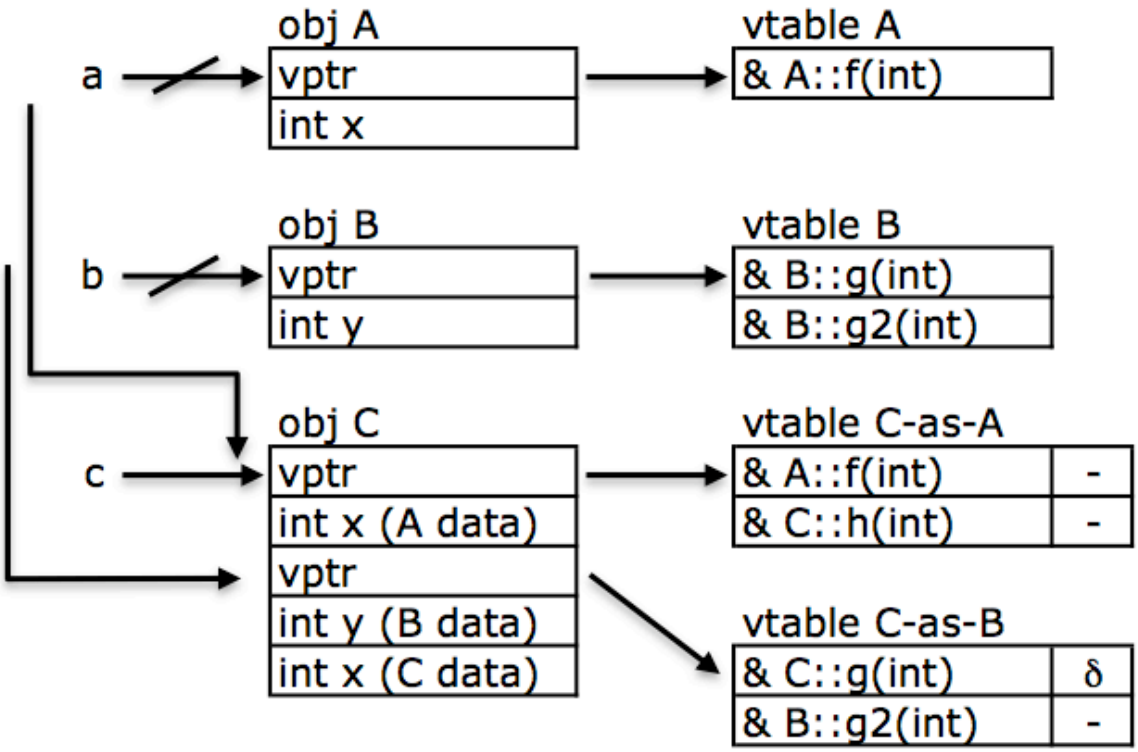
Esercizio 13

```
// esame 5 luglio 2007

#include<iostream>
using namespace std;

class A{
      int x;
    public:
      A(int a=0){x=a;}
      void r(int x) {f();}
      virtual void f() {cout<<"f di A"<<endl;}
      virtual void g() {cout<<"g di A"<<endl;}
    };

class B{
      int y;
    public:
      B(int a=0){y=a;}
      virtual void g(){cout<<"g di B"<<endl;}
      virtual void h(){k(y);cout<<"h di B"<<endl;}
      void k(int a){cout<<a<<endl; h();}
    };

class C: public A, public B{
    public:
      int z;
      C(int a=0, int b=0, int c=0):A(a), B(b){z=c;}
      virtual void g(){cout<<"g di C"<<endl;}
      virtual void s(){cout<<"s di C"<<endl;}
      virtual void h(){cout<<"h di C"<<endl;f();}
    };

main(){

A*a; B*b; C*c=new C(1,2,3);
//B* b2= new B(2);
//b2->k(10);      //stamperebbe 10 una volta e 2 all'infinito
a=c;
b=c;
c->k(1);          // 1, h di C, f di A
a->g();           // g di C
b->g();           // g di C
//b->f();         // non compila, in b non c'e' f()
b->k(10);         // 10, h di C, f di A

}
```


obj A
a
vptr
int x
vtable A
& A::f()
& A::g()
obj B
b
vptr
int y
vtable B
& B::g()
& B::h()
obj C
c
vptr
int x (A data)
vptr
int y (B data)
int x (C data)
vtable C-as-A
& A::f()
-
& C::g()
0
& C::s(int)
-
vtable C-as-B
& C::g()
δ
& C::h()
–

Esercizio 14

```
#include<iostream>
using namespace std;

class A{
    public:
      int x;
      A(int a=0){x=a;}
      virtual void f(int value){
            cout<<"f di A con "<< value <<" - "<< x <<endl;}};

class B{
    public:
      int y;
      B(int b=0){y=b;}
      virtual void g(int value){
            cout<<"g di B con "<< value <<" - "<< y <<endl;}
      virtual void g2(int value){
            cout<<"g2 di B con "<< value <<" - "<< y <<endl;}};

class C: public A, public B{
    public:
      int z;
      C(int a=0, int b=0, int c=0):A(a), B(b){z=c;}
      virtual void g(int value){
            cout<<"g di C con "<< value <<" - "<< y <<endl;}
      virtual void h(int value){
            cout<<"h di C con "<< value <<" - "<< z <<endl;}};

main()
{
  A* a = new A(10);
  B* b = new B(20);
  C* c = new C(1,2,3);
  a->f(42);       //f di A con 42 - 10
  b->g(42);       //g di B con 42 - 20
  b->g2(42);      //g2 di B con 42 - 20
  c->f(42);       //f di A con 42 - 1
  c->g(42);       //g di C con 42 – 2
  c->g2(42);      //g2 di B con 42 – 2
  c->h(42);       //h di C con 42 – 3
  a=c;
  b=c;
  a->f(42);       //f di A con 42 – 1
  b->g(42);       //g di C con 42 – 2

}
```

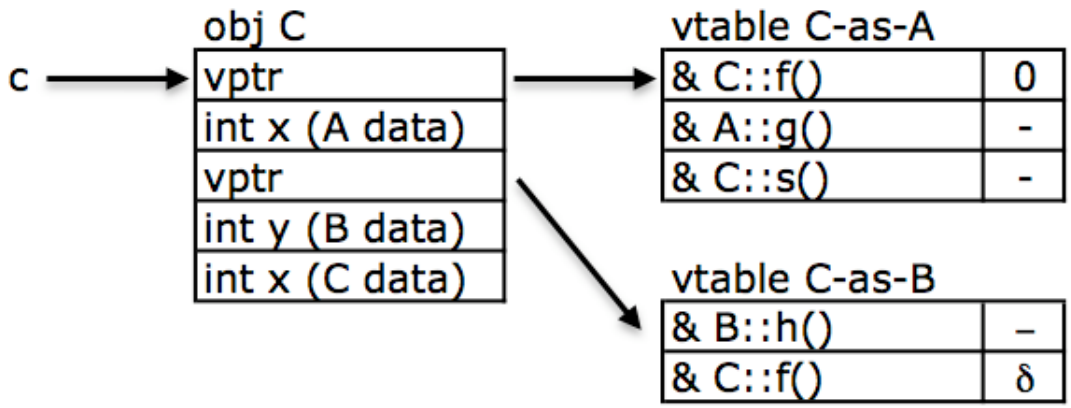

obj A
vtable A
a
vptr
& A::f(int)
int x
obj B
vtable B
b
vptr
& B::g(int)
int y
& B::g2(int)
obj C
vtable C-as-A
c
vptr
& A::f(int)
-
int x (A data)
& C::h(int)
-
vptr
vtable C-as-B
int y (B data)
& C::g(int)
δ
int x (C data)
& B::g2(int)
-

Esercizio 15

```
//esame 4 settembre 2008

#include<iostream>
using namespace std;

class A{
    public:
      int x;
      A(int a=0){x=a;}
      virtual void f(){g();}
      virtual void g(){cout<<"g di A"<<endl;}};

class B{
    public:
      int y;
      B(int b=0){y=b;}
      void r(){f();}
      virtual void h(){cout<<"h di B"<<endl;}
      virtual void f(){h();}};

class C: public A, public B{
    public:
      int z;
      C(int a=0, int b=0, int c=0):A(a), B(b){z=c;}
      virtual void f(){h();}
      virtual void s(){r();}};
```

a) Si chiede di disegnare in dettaglio la struttura di un oggetto della classe C spiegando le ragioni di questa struttura.
b) Si chiede di spiegare il tipo dei parametri formali dei metodi: s, r, C::f, B::f, e B::h e anche di descrivere come vengono compilate le invocazioni contenute nel corpo di questi 5 metodi.
c) Si consideri il seguente frammento di programma: C*c=newC(1,2,3); c->s();Si chiede si specificare la sequenza di invocazioni a metodi che viene causata dall'invocazione di c-> s(). Per ogni metodo invocato spiegare se viene raggiunto passando per la prima o per la seconda vtabledi c e perché. Questa risposta deve essere praticamente già contenuta nella risposta (b).

Breve soluzione



signature s(C*o)
o->r()
r(B*o)
viene passato a r l'o-delta

signature r(B*o)
o->f()
f(B*o)

signature C::f(C*o)
if (displ==0) *(o+delta->vptr[1])(o+delta) else *(o->vptr[1])(o)

signature B::f(B*o)
*(o->vptr[1])(o)

C*c=newC(1,2,3);
c->s();

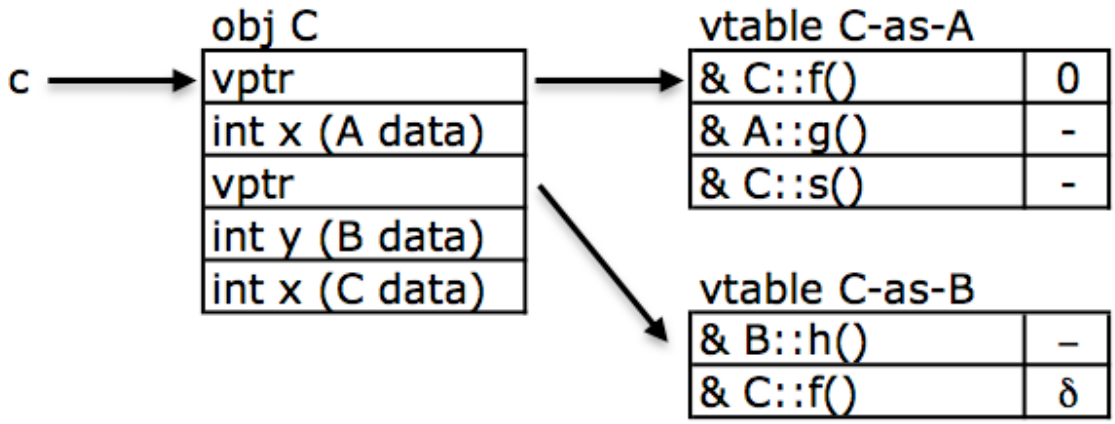
s(C*o)
*(o->vptr[3])(o)
r(B*o)
viene passato come o, o+delta
o->f()
f(C*o)
siamo entrati da displ!=0 cioè dalla vtable di B
quindi si farà *(o->vptr[1])(o)
per l'invocazione di h che stamperà "h di B"

Esercizio 16

```
//esame del 17 marzo 2008

#include<iostream>
using namespace std;

class A{
public:
    int x;
    A(int a=0){x=a;}
    void r(){f();}
    virtual void f(){g();}
    virtual void g(){cout<< 2<<x<<endl;}};

class B{
public:
    int y;
    B(int b=0){y=b;}
    virtual void h(){cout<<"h di B"<<endl;}
    virtual void f(){cout<<3<<endl;}};

class C: public A, public B{
public:
    int z;
    C(int a=0, int b=0, int c=0):A(a),B(b){z=c;}
    virtual void f(){h();}
    virtual void s() {r();}
};
```

a) Si chiede di disegnare in dettaglio la struttura di un oggetto della classe C spiegando le ragioni di questa struttura.
b) Si consideri ora il seguente frammento di programma che usa le precedenti classi:
A*a; B* b; C* c=new C(1,2,3);
a=c;
b=c;
qual'è il rapporto tra i valori di a , b e c dopo queste assegnazioni?
c) Come verrebbe compilata l'invocazione di f nel corpo di r? E come verrebbe compilata l'invocazione di h nel corpo di f della classe C?
d) Spiegare la sequenza di invocazioni a metodi che viene prodotta dall'invocazione di c-> s(). Per ogni metodo spiegare se viene raggiunto passando per la prima o per la seconda vtable di c e perché. Usate la vostra risposta a (c) per spiegare quello che succede.

Breve soluzione



invocazione di f nel corpo di r
r(A*o)
o->f()
f(A*o)
*(o->vptr[1])(o)

invocazione di h nel corpo di C::f
f(C*o)
if (displ==0) *(o+delta->vptr[1])(o+delta) else *(o->vptr[1])(o)

sequenza di invocazioni di c->s()
s(C*o)
o->r()
r(A*o)
A* è uguale a C* quindi il puntatore non cambia nel passaggio del parametro
Esegue l'invocazione di f nel corpo di r e poi h nel corpo di C::f per stampare "h di B"

Esercizio 17

```
//esame del 7 luglio 2008

#include<iostream>
using namespace std;

class A{
public:
    int a;
    A(int x=0){a=x;}
    virtual int a1(){return a+1;}
    virtual int a2(){return a+2;}
};

class B{
public:
    int b;
    B(int x=0){b=x;}
    virtual int b1(){return b+1;}
    virtual int b2(){return b+2;}
    int g(){return b1()+2;}
};

class C: public A,public B {
public:
    int c;
    C(int x=0,int y=0,int z=0):A(x),B(y){c=z;}
    virtual int b1(){return a1()+b2() +a;}
    virtual int c1(){return g()+a2() ;}
};
```

a) disegnare la struttura di un oggetto della classe C, spiegando (brevemente) le motivazioni principali di questa struttura.

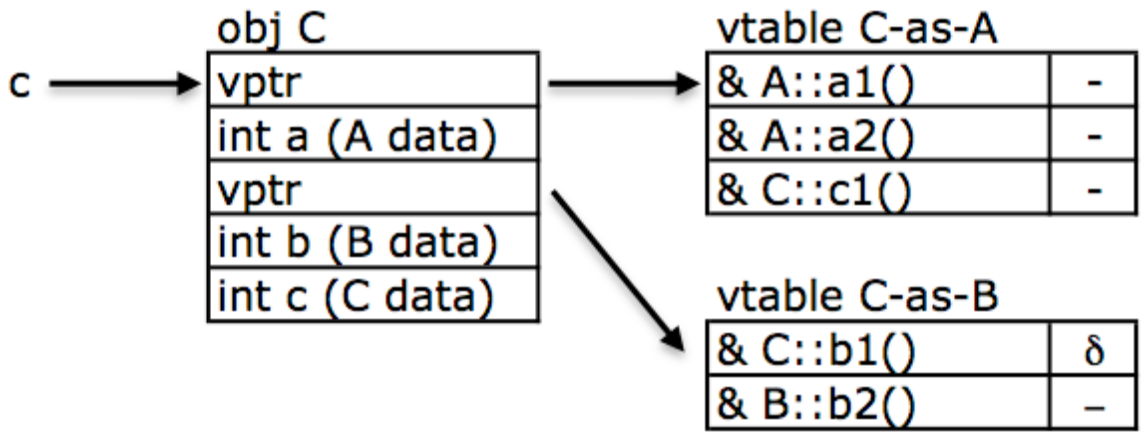
b) Considerare il metodo B::g() ed i metodi C::b1() e C::c1() e per ciascuno di essi specificare il tipo del parametro implicito (destinato ad accogliere l'oggetto d'invocazione del metodo) e la traduzione che il compilatore fa delle invocazioni di altri metodi contenute nel corpo di questi metodi. In particolare, per il metodo C::b1() dovete spiegare il tipo che il compilatore assume per l'oggetto d'invocazione e spiegare come il compilatore traduce le invocazioni a1() e b2() e anche come traduce l'accesso al campo a.

c) Si consideri il seguente programma C++ che usa le classi A, B e C definite prima:

```
main()
{
        B* pb;
        C* pc=new C(3,4,5);
        pb=pc;
        pb->b1(); // (i)
        pc->b1(); // (ii)
}
```

Spiegare come il compilatore traduce le invocazioni (i) e (ii) . In funzione di questa risposta spiegare se queste 2 invocazioni calcolano valori uguali o diversi e quale/quali valori calcolano. Infine per ciascuna invocazione specificare quale sequenza di invocazioni di metodi essa causa e per ciascun metodo invocato, indicare quale è il tipo dinamico del suo oggetto di invocazione e perché e anche attraverso quale virtual table il metodo viene raggiunto. Osservate che i tipi dinamici che indicate devono essere coerenti con quelli statici indicati nella risposta (b).

Breve soluzione



signature B::g(B*o)
o->b1()
b1(B*o)
*(o->vptr[1])(o)

signature C::b1(C*o)
invoca a1() e b2()

if (displ==0)

o->a1()
a1(A*o)
C* come attuale che è uguale a A*
*(o->vptr[1])(o)

o->b2()
b2(B*o)
C* come attuale
*(o+delta->vptr[2])(o+delta)

if (displ!=0)

o->a1()
a1(A*o)
*(o-delta->vptr[1])(o-delta)

o->b2()
b2(B*o)
C* come attuale
*(o->vptr[2])(o)

signature C::c1(C*o)
invoca g() e a2()

o->g()
g(B*o)
viene passato a g o+delta
o->a2()
a2(A*o)
C* come attuale che è uguale a A*
*(o->vptr[2])(o)

il main stampa 13 13

Esercizio 18

```
//private virtual es 12.7 del libro

#include<iostream>
using namespace std;

class A {
    public:
      void f(){g();}
    private:
      virtual void g(){cout<<'A'<<endl;}
};

class B: public A {
    private:
      virtual void g(){cout<<'B'<<endl;}};

main()
{
    A* pa=new A, *aa; B* pb=new B;
    pa->f();
    pb->f();
    aa=pb;
    aa->f();
}


// stampa A B B
```

Esercizio 19

```
//esame 4 settembre 2007

#include<iostream>
using namespace std;

class A{
    int x;
    public: A(int b=0){x=a;}
      virtual void f(){cout<<"f di A"<<endl;}
      virtual void g(){cout<<"g di A"<<endl;}};

class B{
    int y;
    public: B(int b=0){y=b;}
      virtual void g(){cout<<"g di B"<<endl;}
      virtual void h(){cout<<"h di B"<<endl;}
      void d(){cout<<"d di B"<<endl; g();}};

class C: public A, public B{
    int z;
    public: C(int a=0, int b=0, int c=0):A(a),B(b){z=c;}
      virtual void g(){cout<<"g di C"<<endl; f();}
      virtual void h(){cout<<"h di C"<<endl;}};
```

a) Descrivere la struttura di oggetti di tipo A, B e C; Descrivere inoltre il prototipo del metodo g della classe C, del metodo f della A, e del metodo d della classe B.specificare inoltre come vengono compilate le invocazioni ad altri metodi contenute in C::g e B::d
b) Si consideri ora le seguenti istruzioni che usano le precedenti dichiarazioni
   A* a; B* b; C* c=new C(1,2,3); b=c; a=c; // (1)
   b->d(); // (2)
   a->g(); // (3)
   Si chiede di:
   i. spiegare la relazione tra i valori di c, b ed a dopo le istruzioni (1)
   ii. descrivere quale sequenza di invocazioni di metodi sono causate dalle invocazioni (2) e (3), specificando, per ogni metodo invocato, attraverso quale delle 2 virtual tables di c esso è invocato e com'è possibile che questo succeda (la spiegazione deve far uso della compilazione descritta al punto (a)).

Esercizio 20

```
//esame 12 settembre 2007

#include<iostream>
using namespace std;

class A{
    int x;
    public:
      A(int a=0){x=a;}
      virtual void f(){cout<<"f di A"<<endl;}
      virtual void g(){cout<<"g di A"<<endl;}};

class B{
    int y;
    public:
      B(int a=0){y=a;}
      virtual void g(){cout<<"g di B"<<endl;}
      virtual void h(){cout<<"h di B"<<endl;}};

class C: public A, public B{
    int z;
    public:
      C(int a=0, int b=0, int c=0):A(a),B(b){z=c;}
      virtual void g(){cout<<"g di C"<<endl; f();}
            virtual void s(){cout<<" s di C"<<endl; h();}};
```

Descrivere la struttura di oggetti di tipo A, B e C; Per i metodi g() ed s() di C, specificare quali sono i loro parametri formali e come il compilatore traduce l'invocazione in essi contenuta.