# COSTRUZIONE -> LIGHTTOSI (C)

#### Esercizio Cosa Stampa

```
class Z
                                                                                                               class D: virtual public B {
 public: Z(int x) {}
                                                                                                                  D* f(Z) {cout << "D::f(Z) "; f(3.14); return this;}
                                                  F* puntF=newF;
                                                                                                                  virtual void f(double) {cout << "D::f(double) ";}
D() {cout << "D() ";}</pre>
class A {
public:
                                                                                                                   ~D() {cout << "~D ";}
  A() {cout << "A() "; }
~A() {cout << "~A ";}
                                                                                                               class E: public C {
                                                                                                                 unite:
virtual void f() {cout << "E::f() "; C::f(Z(1));}
C* f(Z) {cout << "E::f(Z) "; f(); return this;}
E() {cout << "E() "; }
E(const E& e) {cout << "Ec ";}</pre>
class B: public A {
  public:
    void f(int) {cout << "B::f(int) "; f(3.14); }
    virtual void f(double) {cout << "B::f(double) ";}
    virtual B* f(Z) {cout << "B::f(Z) "; return this; }
    B() {cout << "B() "; }</pre>
                                                                                                                  E(const E& e) {cout < ~~E() {cout < ~~~E";}
    ~B() {cout << "~B ";}
                                                    A() B() (() D()
                                                                                                               class F: public E, public D {
                                                                                                               public:
  void f() const {cout << "F::f() ";}</pre>
class C: virtual public B {
                                                                             5 () F()
                                                                                                                  Void f() const {cout << "F::f() ";
F* f(Z) {cout << "F::f(Z) "; return this;
void f(double) {cout << "F::f(double) ";}
F() {cout << "F() ";
}
F() {cout << "F";}</pre>
 public:
  virtual void f(const int&) {cout<< "C::f(const int&) ";}
virtual C* f(Z) {cout << "C::f(Z) "; return this;}
C() {cout << "C() "; }
virtual ~C() {cout << "~C ";}</pre>
A* pa = new F; D* pd = new D; E* pe = new E; F* pf = new F; B *pbl=pd, *pb3=pf; C* pc=pf;
```

## COPIA Esercizio Cosa Stampa class Z { class D: virtual public B { public: Z(int x) {} public: $E^*$ puntE = new E(\*pe);DBD16: D\* f(Z) {cout << "D::f(Z) "; f(3.14); return this;} virtual void f(double) {cout << "D::f(double) ";} D() {cout << "D() ";} D() {cout << "^D ";}</pre> public: A() {cout << "A() "; } ~A() {cout << "~A ";}</pre> }: class E: public C { public: virtual void f() {cout << "E::f() "; C::f(Z(1));} C\* f(Z) {cout << "E::f(Z) "; f(); return this;}</pre> class B: public A { public: void f(int) {cout << "B::f(int) "; f(3.14); } virtual void f(double) {cout << "B::f(double) ";} virtual B\* f(Z) {cout << "B::f(Z) "; return this; } B() {cout << "B() "; } B() {cout << "B"; } </pre> }; class F: public E, public D { public: void f() const {cout << "F::f() ";}</pre> F\* f(Z) {cout << "F::f(Z) "; return this;} void f(double) {cout << "F::f(double) ";} virtual void f(const int&) {cout<< "C::f(const int&) ";}</pre> F() {cout << "F() "; } F() {cout << "F"; } virtual C\* f(2) {cout << "C::f(Z) "; return this;} C() {cout << "C() "; } virtual C() {cout << "C";}</pre> A\* pa = new F; D\* pd = new D; E\* pe = new E; F\* pf = new F; B \*pb1=pd, \*pb3=pf; C\* pc=pf;

# N.B!

# Esercizio Cosa Stampa

```
class D: virtual public B {
 public: Z(int x) {}
                                                                                                 public:
                                                                                                    Dat f(Z) {cout << "D::f(Z) "; f(3.14); return this;}
virtual void f(double) {cout << "D::f(double) ";}
D() {cout << "D() ";}</pre>
class A {
public:
    A() {cout << "A() "; }
    ~A() {cout << "~A ";}</pre>
                                                                                                     ~D() {cout << "~D ";}
                                                                                                 class E: public C {
                                                                                                 public:
                                                                                                    class B: public A {
 public:
  public:
    void f(int) {cout << "B::f(int) "; f(3.14); }
    virtual void f(double) {cout << "B::f(double) ";}
    virtual B* f(Z) {cout << "B::f(Z) "; return this;
    B() {cout << "B() "; }
    ~B() {cout << "B";}</pre>
                                                                                                                                                FOR OPONATIONZ ()
class C: virtual public B {
public:
virtual:
                                                                                                 class F: public E, public D {
public:
                                                                                                                                                         D: : Operno e c
                                                                                                    ublic:
void f() const {cout << "F::f() ";)
F* f(Z) {cout << "F::f(Z) "; return this;}
void f(double) {cout << "F::f(double) ";}
F() {cout << "F() ";}
-F() {cout << "F() ";}</pre>
                                                                  D() (5() F()
   virtual void f(const int&) {cout<< "C::f(const int&) ";}</pre>
  virtual C* f(Z) {cout << "C::f(Z) "; return this;}
C() {cout << "C() "; }
virtual "C() {cout << ""C ";}</pre>
                                                                                                                                                                    SCANSAND
A* pa = new F; D* pd = new D; E* pe = new E; F* pf = new F; B *pb1=pd, *pb3=pf; C* pc=pf;
```

#### Esercizio Cosa Stampa

```
class D: virtual public B {
 public: Z(int x) {}
                                                                                                      public:
                                                  pb3->f(3);
                                                                                                        D* f(Z) {cout << "D::f(Z) "; f(3.14); return this;} virtual void f(double) {cout << "D::f(double) ";} D() {cout << "D() ";}
                                               TS (B)
class A {
public:
                                                                                                         ~D() {cout << "~D ";}
  A() {cout << "A() "; }
    ~A() {cout << "~A ";}
                                                                                                      class E: public C {
                                                                                                      public:
                                                         (Y)
                                                                                                        contended:
contended f() {cout << "E::f() "; C::f(Z(1));}
C* f(Z) {cout << "E::f(Z) "; f(); return this;}
E() {cout << "E() "; }
E(const E& e) {cout << "Ec ";}
~E() {cout << "~E ";}</pre>
class B: public A {
                                                                           6
   void f(int) {cout << "B::f(int) "; f(3.14); }
virtual void f(double) {cout << "B::f(double) ";}</pre>
 public:
                                                                                   5
   virtual B* f(Z) {cout << "B::f(Z) "; return this; }
B() {cout << "B() "; }</pre>
    B() {cout << "~B ";}
                                                                                                     class F: public E, public D {
                                                                                                     public:
                                                                                                         void f() const {cout << "F::f() ";}
F* f(Z) {cout << "F::f(Z) "; return this;}</pre>
class C: virtual public B {
 public:
  virtual void f(const int&) {cout<< "C::f(const int&) ";}
virtual C* f(Z) {cout << "C::f(Z) "; return this;}
C() {cout << "C() "; }
virtual ~C() {cout << "~C ";}</pre>
                                                                                                         void f(double) {cout << "F::f(double) ";}
F() {cout << "F() ";}</pre>
                                                                                                          ~F() {cout << "~F ";}
A* pa = new F; D* pd = new D; E* pe = new E; F* pf = new F; B *pb1=pd, *pb3=pf; C* pc=pf;
```

## Esercizio Cosa Stampa

```
class D: virtual public B {
                                                                                          public:
  public: Z(int x) {}
                                        pa->f(1.2):
                                                                                            D* f(Z) {cout << "D::f(Z) "; f(3.14); return this;}
                                                                           A
                                                                                            virtual void f(double) {cout << "D::f(double) ";}
D() {cout << "D() ";}</pre>
 class A {
                                                                                             ~D() {cout << "~D ";}
 public:
   A() {cout << "A() "; } ~A() {cout << "~A ";}
                                                                  B
                                                                                          class E: public C {
                                                                                          public:
                                                                                            Dilic:
virtual void f() {cout << "E::f() "; C::f(Z(1));}
C* f(Z) {cout << "E::f(Z) "; f(); return this;}
E() {cout << "E() "; }
E(const E& e) {cout << "Ec ";}
~E() {cout << "~E ";}</pre>
                                                                          D
 class B: public A {
  public:
   void f(int) {cout << "B::f(int) "; f(3.14);</pre>
                                                                          6
  virtual B* f(Z) {cout << "B::f(Z) "; return this; }
   B() {cout << "B() ";
~B() {cout << "~B ";}
                            "; }
                                                                                          class F: public E, public D {
                                                                                          public:
  void f() const {cout << "F::f() ";}</pre>
 class C: virtual public B {
  public:
                                                                                             F* f(Z) {cout << "F::f(Z) "; return this;}
   virtual void f(const int&) {cout<< "C::f(const int&) ";}
virtual C* f(Z) {cout << "C::f(Z) "; return this;}</pre>
                                                                                             void f(double) {cout << "F::f(double) ";}
F() {cout << "F() "; }</pre>
                                                                                             ~F() {cout << "~F ";}
  C() {cout << "C() "; }
virtual ~C() {cout << "~C";}</pre>
A* pa = new F; D* pd = new D; E* pe = new E; F* pf = new F; B *pb1=pd, *pb3=pf; C* pc=pf;
```

# Esercizio Cosa Stampa

```
FUMBAG (IM) ->
                                                                            oporator (NFC)
class 7 {
                                                                                          class D: virtual public B {
 public: Z(int x) {}
                                                                                          public:
                                                                                             D \star f(Z) {cout << "D::f(Z) "; f(3.14); return this;}
                                                                                             virtual void f(double) {cout << "D::f(double) ";}
D() {cout << "D() ";}</pre>
class A {
public:
                                                                                              ~D() {cout << "~D ";}
                                                                                                                                                      2
  A() {cout << "A() "; } ~A() {cout << "~A ";}
                                                                  B
                                                                                          class E: public C {
                                                                                          public:
                                                                          D
                                                                                             virtual void f() {cout << "E::f() "; C::f(Z(1));}
C* f(Z) {cout << "E::f(Z) "; f(); return this;}
E() {cout << "E() "; }</pre>
class B: public A {
  void f(int) {cout << "B::f(int) "; f(3.14); }
virtual void f(double) {cout << "B::f(double) ";} 5</pre>
                                                                                             E(const E& e) {cout << "Ec ";}
~E() {cout << "~E ";}
  virtual B* f(Z) {cout << "B::f(Z) "; return this; }
B() {cout << "B() "; }
"B() {cout << ""B ";}</pre>
                                                                                            }:
};
                                                                                          class F: public E, public D {
                                                                                          public:
                                                                                             void f() const {cout << "F::f() ";}
F* f(Z) {cout << "F::f(Z) "; return this;}
void f(double) {cout << "F::f(double) ";}</pre>
class C: virtual public B {
public:
  virtual void f(const int&) {cout<< "C::f(const int&) ";}</pre>
  virtual C* f(Z) {cout << "C::f(Z) "; return this;}</pre>
                                                                                             F() {cout << "F() "; }
~F() {cout << "~F";}
  C() {cout << "C() "; }
virtual ~C() {cout << "~C";}</pre>
A* pa = new F; D* pd = new D; E* pe = new E; F* pf = new F; B *pb1=pd, *pb3=pf; C* pc=pf;
```

Esercizio Cosa Stampa

```
if(typeid(pb3)==typeid(F)) pb3->f(Z(2));
                                                                                              class D: virtual public B {
 public: Z(int x) {}
                                                                                              public:
                                              Fact S
                                                                                                class A {
                                                                                                  D() {cout << "~D ";}
public:
  A() {cout << "A() "; } ~A() {cout << "~A ";}
                                                                     B
                                                                                              };
                                                                                              class E: public C {
                                                                             D
                                                                                                virtual void f() {cout << "E::f() "; C::f(Z(1));}
C* f(Z) {cout << "E::f(Z) "; f(); return this;}
E() {cout << "E() "; }
E(const E& e) {cout << "EC ";}</pre>
class B: public A {
   void f(int) {cout << "B::f(int) "; f(3.14); }</pre>
                                                                            6,
  virtual void f(double) {cout << "B::f(double) ";}
virtual B* f(Z) {cout << "B::f(Z) "; return this; }
B() {cout << "B() "; }</pre>
                                                                                                 E(const E& e) {cout < ~~E() {cout << ~~E";}
   B() {cout << "B() ";
~B() {cout << "~B ";}
                                                                                                                                                               F::=(7)
                                                                                              class F: public E, public D {
                                                                                              public:
                                                                                                void f() const {cout << "F::f() ";}
F* f(Z) {cout << "F::f(Z) "; return this;}
void f(double) {cout << "F::f(double) ";}
F() {cout << "F() ";}</pre>
class C: virtual public B {
 public:
  virtual void f(const int&) {cout<< "C::f(const int&) ";}
virtual C* f(Z) {cout << "C::f(Z) "; return this;}</pre>
  C() {cout << "C() "; }
virtual ~C() {cout << "~C";}
                                                                                                  ~F() {cout << "~F ";}
A* pa = new F; D* pd = new D; E* pe = new E; F* pf = new F; \frac{B}{D} *pb1=pd, *pb3=pf; C* pc=pf;
```

```
CAPC = NEW FC>:
                                                                                                                    -> 5 x PC = NOW FO).
Esercizio Cosa Stampa
 class 7 {
                                                                                                       class D: virtual public B {
  public: Z(int x) {}
                                          static_cast<E*>(pc)->f();
                                                                                                       public:
                                                                                                          class A {
 public:
  A() {cout << "A() "; }
~A() {cout << "~A ";}
                                                                            B
                                                                                                       };
                                                                                                       class E: public C {
public:
                                                                                                                                                        6
                                                                                                          united
virtual void f() {cout << "E::f() "; C::f(Z(1));}
C* f(Z) {cout << "E::f(Z) "; f(); return this;}
E() {cout << "E() ";}
E(const E& e) {cout << "Ec ";}
~E() {cout << "E ";}</pre>
                                                                                     D
 class B: public A {
                                                                             0
  public:
   public:
    void f(int) {cout << "B::f(int) "; f(3.14); }
    virtual void f(double) {cout << "B::f(double) ";}
    virtual B* f(Z) {cout << "B::f(Z) "; return this; }
    B() {cout << "B() "; }
    B() {cout << "B"; }
</pre>
                                                                                    5
                                                                                                       - };
                                                                                                       class F: public E, public D {
                                                                                                           void f() const {cout << "F::f() ";}</pre>
 class C: virtual public B {
                                                                                                          void f() cout << "F::f(Z) "; return this;}
void f(double) {cout << "F::f(double) ";}
F() {cout << "F() ";}
^F() {cout << "F";}</pre>
  public:
   virtual void f(const int&) {cout<< "C::f(const int&) ";}
virtual C* f(Z) {cout << "C::f(Z) "; return this;}
C() {cout << "C() "; }
virtual C() {cout << "C" C";}</pre>
 A* pa = new F; D* pd = new D; E* pe = new E; F* pf = new F; B *pb1=pd, *pb3=pf; C* pc=pf;
```

```
B > F(4) -> B :: F(Wr)
class 7 (
                                                                                                                  class D: virtual public B {
 public: Z(int x) {}
                                                                                                                 public:
                                                                                                                    Dutlic:
    D* f(Z) {cout << "D::f(Z) "; f(3.14); return this;}
virtual void f(double) {cout << "D::f(double) ";}
    D() {cout << "D() ";}
    TD() {cout << "^D ";}</pre>
                                                  (pb3->f(Z(3)))->f(4);
class A {
public:
  A() {cout << "A() "; } ~A() {cout << "~A ";}
                                                                                   B
                                                                                                                 };
};
                                                                                                                 class E: public C {
                                                                                                                 public:
                                                                                             D
                                                                                                                    virtual void f() {cout << "E::f() "; C::f(Z(1));}
C* f(Z) {cout << "E::f(Z) "; f(); return this;}
E() {cout << "E() "; }
E(const E& e) {cout << "Ec ";}</pre>
class B: public A {
                                                                                    C
 public:
   void f(int) {cout << "B::f(int) "; f(3.14); }
virtual void f(double) {cout << "B::f(double) ";}</pre>
                                                                                            6
                                                                                                                     E(const E& e) {cout < ~~E() {cout < ~~E";}
   virtual B* f(Z) {cout << "B::f(Z) "; return this; }
B() {cout << "B() "; }
"B() {cout << ""B"; }</pre>
                                                                                                                   };
                                                                                                                                                                                 0
                                                                                                                 class F: public E, public D {
                                                                                                                 public:
                                                                                                                                                                                                     POPULA
                                                                                                                    bblic:
    void f() const {cout << "F::f() ";}
    F* f(Z) {cout << "F::f(Z) "; return t|
    void f(double) {cout << "F::f(double)
    F() {cout << "F() ";}</pre>
class C: virtual public B {
                                                                                                                                                                                                         IN B
 public:
                                                                                                                                                                        return this; }
f(double) "; }
  virtual void f(const int&) {cout<< "C::f(const int&) ";}
virtual C* f(Z) {cout << "C::f(Z) "; return this;}</pre>
                                                                                                                                                                                                        (RASB)
  C() {cout << "C() "; }
virtual ~C() {cout << "~C";}
                                                                                                                      F() {cout << "~F ";}
```

A\* pa = new F; D\* pd = new D; E\* pe = new E; F\* pf = new F; B \*pb1=pd, \*pb3=pf; C\* pc=pf;

B