① COSA-STAMPA

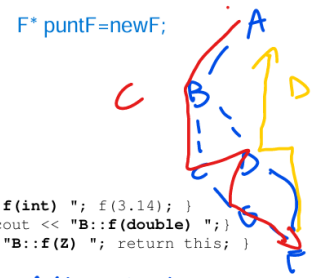
COSTRUZIONE → LEFTMOST (C)

DISTRUZIONE → RIGHTMOST (D)

Esercizio Cosa Stampa

```
class Z {
 public: Z(int x) {}
};

class A {
public:
  A() {cout << "A() "; }
  ~A() {cout << "~A ";}
};

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 ";}
};

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 ";}
};

class D: virtual public B {
public:
  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 ";}
};

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;}
  E() {cout << "E() "; }
  E(const E& e) {cout << "Ec ";}
  ~E() {cout << "~E ";}
 };

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() "; }
  ~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;
```

F* puntF=newF;



A() B() C() D() E() F()

Esercizio Cosa Stampa

COPIA

E* puntE = new E(*pe);

A() B() C() E C

ERR

```
class Z {
 public: Z(int x) {}
};

class A {
public:
  A() {cout << "A() "; }
  ~A() {cout << "~A ";}
};

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 ";}
};

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 ";}
};

class D: virtual public B {
public:
  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 ";}
};

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;}
  E() {cout << "E() "; }
  E(const E& e) {cout << "Ec ";}
  ~E() {cout << "~E ";}
 };

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() "; }
  ~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;
```

N.B!

Esercizio Cosa Stampa

F → A() B() C() D() B() F()

FOR OPERATOR= ()

D:: OPERATOR=

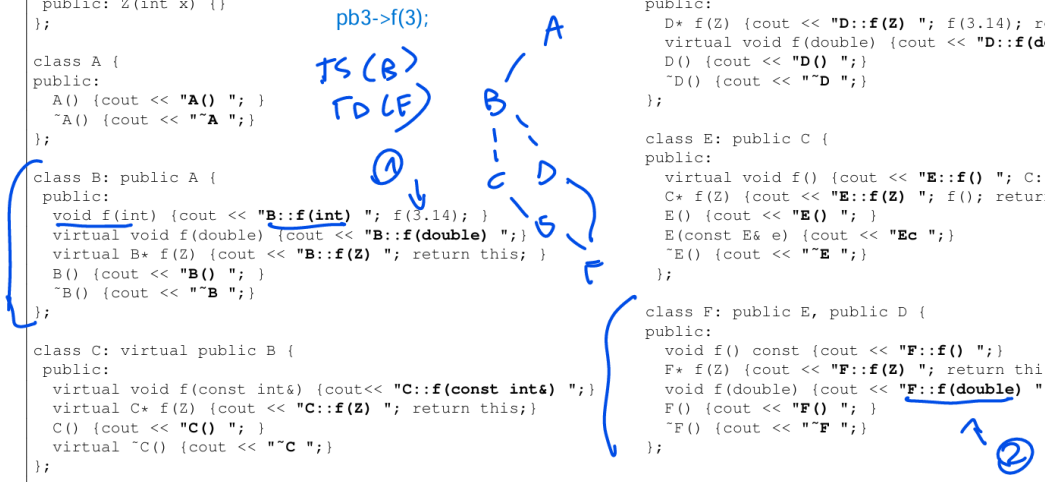
B:: OPERATOR=

STANDARD

```
class Z {
 public: Z(int x) {}
};

class A {
public:
  A() {cout << "A() "; }
  ~A() {cout << "~A ";}
};

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 ";}
};

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 ";}
};

class D: virtual public B {
public:
  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 ";}
};

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;}
  E() {cout << "E() "; }
  E(const E& e) {cout << "Ec ";}
  ~E() {cout << "~E ";}
 };

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() "; }
  ~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;
```

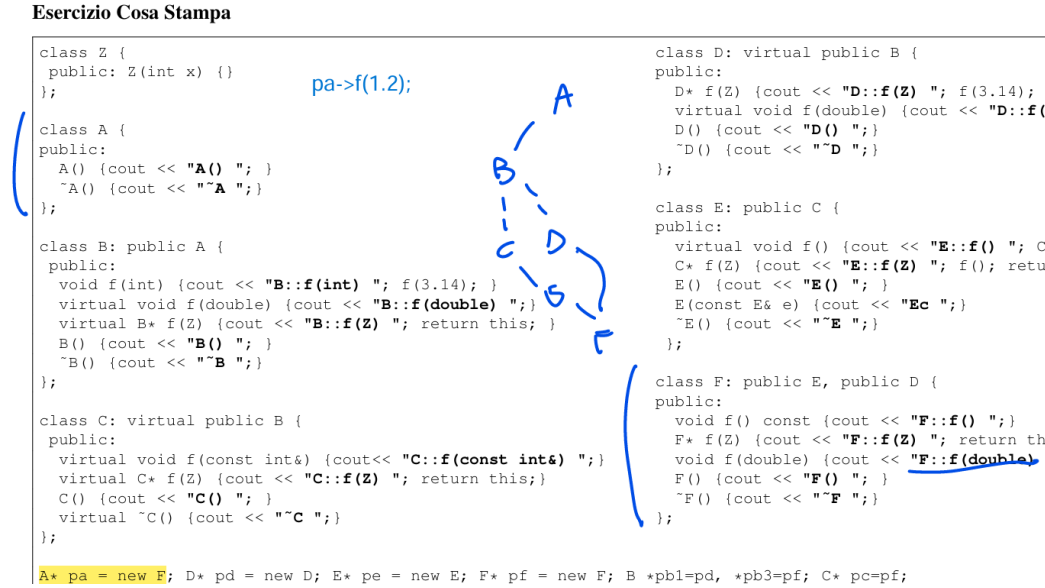
Esercizio Cosa Stampa

```
class Z {
 public: Z(int x) {}
};

class A {
public:
  A() {cout << "A() "; }
  ~A() {cout << "~A ";}
};

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 ";}
};

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 ";}
};

class D: virtual public B {
public:
  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 ";}
};

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;}
  E() {cout << "E() "; }
  E(const E& e) {cout << "Ec ";}
  ~E() {cout << "~E ";}
 };

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() "; }
  ~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;
```



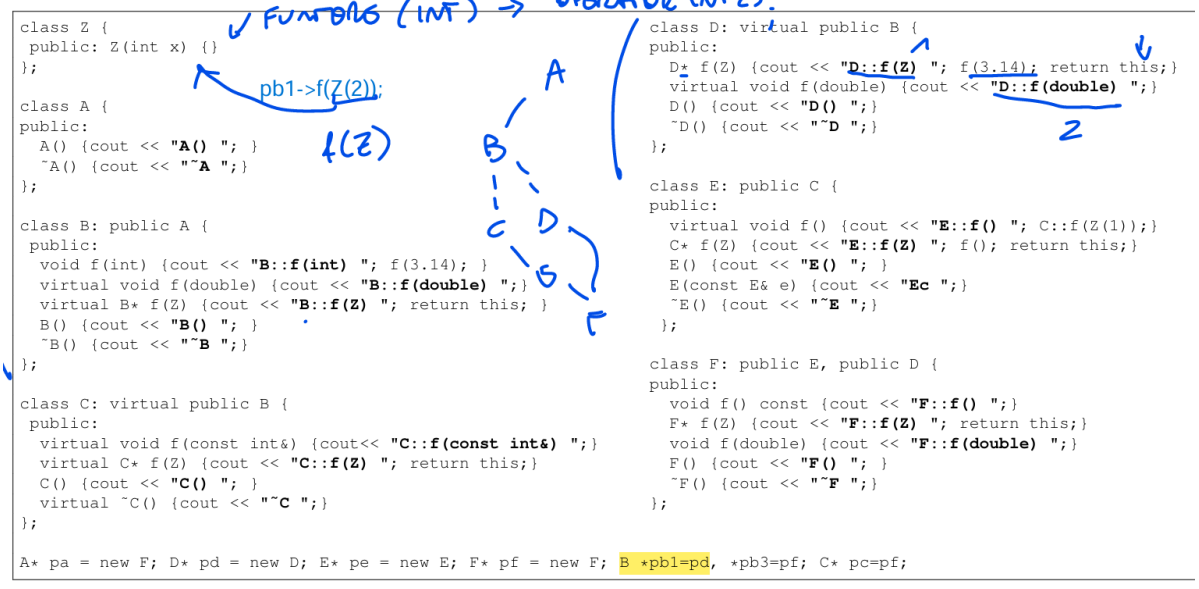
Esercizio Cosa Stampa

```
class Z {
 public: Z(int x) {}
};

class A {
public:
  A() {cout << "A() "; }
  ~A() {cout << "~A ";}
};

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 ";}
};

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 ";}
};

class D: virtual public B {
public:
  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 ";}
};

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;}
  E() {cout << "E() "; }
  E(const E& e) {cout << "Ec ";}
  ~E() {cout << "~E ";}
 };

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() "; }
  ~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;
```



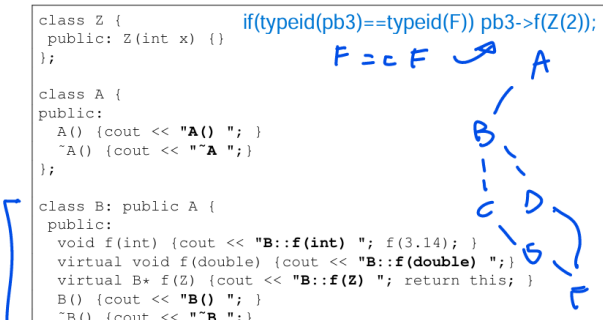
Esercizio Cosa Stampa

```
class Z {
 public: Z(int x) {}
};

class A {
public:
  A() {cout << "A() "; }
  ~A() {cout << "~A ";}
};

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 ";}
};

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 ";}
};

class D: virtual public B {
public:
  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 ";}
};

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;}
  E() {cout << "E() "; }
  E(const E& e) {cout << "Ec ";}
  ~E() {cout << "~E ";}
 };

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() "; }
  ~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;
```

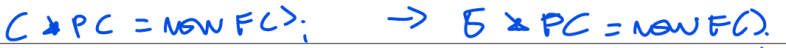


Esercizio Cosa Stampa

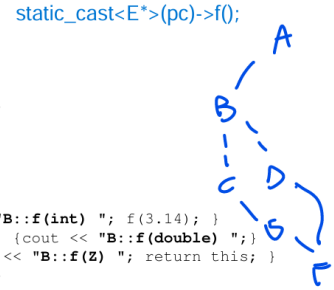
```
class Z {
 public: Z(int x) {}
};

class A {
public:
  A() {cout << "A() "; }
  ~A() {cout << "~A ";}
};

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 ";}
};

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 ";}
};

class D: virtual public B {
public:
  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 ";}
};

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;}
  E() {cout << "E() "; }
  E(const E& e) {cout << "Ec ";}
  ~E() {cout << "~E ";}
 };

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() "; }
  ~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 Z {
 public: Z(int x) {}
};

class A {
public:
  A() {cout << "A() "; }
  ~A() {cout << "~A ";}
};

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 ";}
};

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 ";}
};

class D: virtual public B {
public:
  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 ";}
};

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;}
  E() {cout << "E() "; }
  E(const E& e) {cout << "Ec ";}
  ~E() {cout << "~E ";}
 };

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() "; }
  ~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 Z {
 public: Z(int x) {}
};

class A {
public:
  A() {cout << "A() "; }
  ~A() {cout << "~A ";}
};

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 ";}
};

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 ";}
};

class D: virtual public B {
public:
  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 ";}
};

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;}
  E() {cout << "E() "; }
  E(const E& e) {cout << "Ec ";}
  ~E() {cout << "~E ";}
 };

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() "; }
  ~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;
```