

Laurea in Informatica – Programmazione ad Oggetti – Appello d'Esame 25/01/2024

Nome..... Cognome..... Matricola.....

Esercizio Cosa Stampa

```
class A {
public:
    A() {cout<< " A() ";}
    ~A() {cout<< " ~A ";}
    A(const A& x) {cout<< " Ac ";}
    virtual const A* j() {cout<<" A::j "; return this;}
    virtual void k() {cout <<" A::k "; m();}
    void m() {cout <<" A::m "; j();}
};

class C: virtual public B {
public:
    C() {cout<< " C() ";}
    ~C() {cout<< " ~C ";}
    void g() const {cout <<" C::g ";}
    void k() override {cout <<" C::k "; B::n();}
    virtual void m() {cout <<" C::m "; g(); j();}
    B& n() override {cout <<" C::n "; return *this;}
};

class E: public C, public D {
public:
    E() {cout<< " E() ";}
    ~E() {cout<< " ~E ";}
    E(const E& x) {cout<< " Ec ";}
    virtual void g() const {cout <<" E::g ";}
    const E* j() {cout <<" E::j "; return this;}
    void m() {cout <<" E::m "; g(); j();}
    D& n() final {cout <<" E::n "; return *this;}
};

A* p1 = new E(); B* p2 = new C(); A* p3 = new D(); B* p4 = new E();
const A* p5 = new D(); const B* p6 = new E(); const E* p7 = new E();
```

class B: virtual public A {  
public:  
 B() {cout<< " B() ";}  
 virtual ~B() {cout<< " ~B ";}  
 virtual void g() const {cout <<" B::g ";}  
 virtual const B\* j() {cout <<" B::j "; n(); return this;}  
 void k() {cout <<" B::k "; j(); m();}  
 void m() {cout <<" B::m "; g(); j();}  
 virtual A& n() {cout <<" B::n "; return \*this;}  
};

class D: virtual public B {  
public:  
 D() {cout<< " D() ";}  
 ~D() {cout<< " ~D ";}  
 virtual void g() {cout <<" D::g ";}  
 const B\* j() {cout <<" D::j "; return this;}  
 void k() const {cout <<" D::k "; k();}  
 void m() {cout <<" D::m "; g(); j();}  
};

*NON COMPILA* (NON COMPILE)  
*3 \* (NON COMPILE)*  
(dynamic\_cast<const E\*>(p1->j()))->g();

Le precedenti definizioni compilano correttamente. Per ognuna delle seguenti istruzioni scrivere nell'apposito spazio:

- **NON COMPILA** se la compilazione dell'istruzione provoca un errore;
- **UNDEFINED** se lo statement compila correttamente ma la sua esecuzione provoca un undefined behaviour o un errore run-time;
- se l'istruzione compila correttamente e non provoca errori a run-time allora si scriva la stampa che l'esecuzione produce in output su cout; se non provoca alcuna stampa allora si scriva **NESSUNA STAMPA**

Esercizio Cosa Stampa

```
class A {
public:
    A() {cout<< " A() ";}
    ~A() {cout<< " ~A ";}
    A(const A& x) {cout<< " Ac ";}
    virtual const A* j() {cout<<" A::j "; return this;}
    virtual void k() {cout <<" A::k "; m();}
    void m() {cout <<" A::m "; j();}
};

class C: virtual public B {
public:
    C() {cout<< " C() ";}
    ~C() {cout<< " ~C ";}
    void g() const {cout <<" C::g ";}
    void k() override {cout <<" C::k "; B::n();}
    virtual void m() {cout <<" C::m "; g(); j();}
    B& n() override {cout <<" C::n "; return *this;}
};

class E: public C, public D {
public:
    E() {cout<< " E() ";}
    ~E() {cout<< " ~E ";}
    E(const E& x) {cout<< " Ec ";}
    virtual void g() const {cout <<" E::g ";}
    const E* j() {cout <<" E::j "; return this;}
    void m() {cout <<" E::m "; g(); j();}
    D& n() final {cout <<" E::n "; return *this;}
};

A* p1 = new E(); B* p2 = new C(); A* p3 = new D(); B* p4 = new E();
const A* p5 = new D(); const B* p6 = new E(); const E* p7 = new E();
```

class B: virtual public A {  
public:  
 B() {cout<< " B() ";}  
 virtual ~B() {cout<< " ~B ";}  
 virtual void g() const {cout <<" B::g ";}  
 virtual const B\* j() {cout <<" B::j "; n(); return this;}  
 void k() {cout <<" B::k "; j(); m();}  
 void m() {cout <<" B::m "; g(); j();}  
 virtual A& n() {cout <<" B::n "; return \*this;}  
};

class D: virtual public B {  
public:  
 D() {cout<< " D() ";}  
 ~D() {cout<< " ~D ";}  
 virtual void g() {cout <<" D::g ";}  
 const B\* j() {cout <<" D::j "; return this;}  
 void k() const {cout <<" D::k "; k();}  
 void m() {cout <<" D::m "; g(); j();}  
};

*2*  
*3*  
*1*  
*4*  
*P2*

## Esercizio Cosa Stampa

```
class A {
public:
    A() {cout<< " A() ";}
    ~A() {cout<< " ~A ";}
    A(const A& x) {cout<< " Ac ";}
    virtual const A* j() {cout<< " A::j "; return this;}
    virtual void k() {cout<< " A::k "; m();}
    void m() {cout<< " A::m "; j();}
};
```

```
class C: virtual public B {
public:
    C() {cout<< " C() ";}
    ~C() {cout<< " ~C ";}
    void g() const {cout<< " C::g ";}
    void k() override {cout<< " C::k "; B::n();}
    virtual void m() {cout<< " C::m "; g(); j();}
    B& n() override {cout<< " C::n "; return *this;}
};
```

```
class E: public C, public D {
public:
    E() {cout<< " E() ";}
    ~E() {cout<< " ~E ";}
    E(const E& x) {cout<< " Ec ";}
    virtual void g() const {cout<< " E::g ";}
    const E* j() {cout<< " E::j "; return this;}
    void m() {cout<< " E::m "; g(); j();}
    D& n() final {cout<< " E::n "; return *this;}
};
```

```
A* p1 = new E(); B* p2 = new C(); A* p3 = new D(); B* p4 = new E();
const A* p5 = new D(); const B* p6 = new E(); const E* p7 = new E();
```

```
class B: virtual public A {
public:
    B() {cout<< " B() ";}
    virtual ~B() {cout<< " ~B ";}
    virtual void g() const {cout<< " B::g ";}
    virtual const B* j() {cout<< " B::j "; n(); return this;}
    void k() {cout<< " B::k "; j(); m();}
    void m() {cout<< " B::m "; g(); j();}
    virtual A& n() {cout<< " B::n "; return *this;}
};
```

```
class D: virtual public B {
public:
    D() {cout<< " D() ";}
    ~D() {cout<< " ~D ";}
    virtual void g() {cout<< " D::g ";}
    const B* j() {cout<< " D::j "; return this;}
    void k() const {cout<< " D::k "; k();}
    void m() {cout<< " D::m "; g(); j();}
};
```

P2 → 30 → 60

## Esercizio Cosa Stampa

```
class A {
public:
    A() {cout<< " A() ";}
    ~A() {cout<< " ~A ";}
    A(const A& x) {cout<< " Ac ";}
    virtual const A* j() {cout<< " A::j "; return this;}
    virtual void k() {cout<< " A::k "; m();}
    void m() {cout<< " A::m "; j();}
};
```

```
class C: virtual public B {
public:
    C() {cout<< " C() ";}
    ~C() {cout<< " ~C ";}
    void g() const {cout<< " C::g ";}
    void k() override {cout<< " C::k "; B::n();}
    virtual void m() {cout<< " C::m "; g(); j();}
    B& n() override {cout<< " C::n "; return *this;}
};
```

```
class E: public C, public D {
public:
    E() {cout<< " E() ";}
    ~E() {cout<< " ~E ";}
    E(const E& x) {cout<< " Ec ";}
    virtual void g() const {cout<< " E::g ";}
    const E* j() {cout<< " E::j "; return this;}
    void m() {cout<< " E::m "; g(); j();}
    D& n() final {cout<< " E::n "; return *this;}
};
```

```
A* p1 = new E(); B* p2 = new C(); A* p3 = new D(); B* p4 = new E();
const A* p5 = new D(); const B* p6 = new E(); const E* p7 = new E();
```

```
class B: virtual public A {
public:
    B() {cout<< " B() ";}
    virtual ~B() {cout<< " ~B ";}
    virtual void g() const {cout<< " B::g ";}
    virtual const B* j() {cout<< " B::j "; n(); return this;}
    void k() {cout<< " B::k "; j(); m();}
    void m() {cout<< " B::m "; g(); j();}
    virtual A& n() {cout<< " B::n "; return *this;}
};
```

```
class D: virtual public B {
public:
    D() {cout<< " D() ";}
    ~D() {cout<< " ~D ";}
    virtual void g() {cout<< " D::g ";}
    const B* j() {cout<< " D::j "; return this;}
    void k() const {cout<< " D::k "; k();}
    void m() {cout<< " D::m "; g(); j();}
};
```

P3 → K()

## Esercizio Cosa Stampa

```

class A {
public:
    A() {cout<< " A() ";}
    ~A() {cout<< " ~A ";}
    A(const A& x) {cout<< " Ac ";}
    virtual const A* j() {cout<< " A::j "; return this;}
    virtual void k() {cout<< " A::k "; m();}
    void m() {cout<< " A::m "; j();}
};

class C: virtual public B {
public:
    C() {cout<< " C() ";}
    ~C() {cout<< " ~C ";}
    void g() const {cout<< " C::g ";}
    void k() override {cout<< " C::k "; B::n();}
    virtual void m() {cout<< " C::m "; g(); j();}
    B& n() override {cout<< " C::n "; return *this;}
};

class E: public C, public D {
public:
    E() {cout<< " E() ";}
    ~E() {cout<< " ~E ";}
    E(const E& x) {cout<< " Ec ";}
    virtual void g() const {cout<< " E::g ";}
    const E* j() {cout<< " E::j "; return this;}
    void m() {cout<< " E::m "; g(); j();}
    D& n() final {cout<< " E::n "; return *this;}
};

A* p1 = new E(); B* p2 = new C(); A* p3 = new D(); B* p4 = new E();
const A* p5 = new D(); const B* p6 = new E(); const E* p7 = new E();
    
```

FINAL → OVERRIDING NON VA DUE GIÙ!

## Esercizio Cosa Stampa

```

class A {
public:
    A() {cout<< " A() ";}
    ~A() {cout<< " ~A ";}
    A(const A& x) {cout<< " Ac ";}
    virtual const A* j() {cout<< " A::j "; return this;}
    virtual void k() {cout<< " A::k "; m();}
    void m() {cout<< " A::m "; j();}
};

class C: virtual public B {
public:
    C() {cout<< " C() ";}
    ~C() {cout<< " ~C ";}
    void g() const {cout<< " C::g ";}
    void k() override {cout<< " C::k "; B::n();}
    virtual void m() {cout<< " C::m "; g(); j();}
    B& n() override {cout<< " C::n "; return *this;}
};

class E: public C, public D {
public:
    E() {cout<< " E() ";}
    ~E() {cout<< " ~E ";}
    E(const E& x) {cout<< " Ec ";}
    virtual void g() const {cout<< " E::g ";}
    const E* j() {cout<< " E::j "; return this;}
    void m() {cout<< " E::m "; g(); j();}
    D& n() final {cout<< " E::n "; return *this;}
};

A* p1 = new E(); B* p2 = new C(); A* p3 = new D(); B* p4 = new E();
const A* p5 = new D(); const B* p6 = new E(); const E* p7 = new E();
    
```

((dynamic\_cast<D\*>(p4)) -> n()) . k();  
 D \* p4 = new E() E::n() C::k

## Esercizio Cosa Stampa

```

class A {
public:
    A() {cout<< " A() ";}
    ~A() {cout<< " ~A ";}
    A(const A& x) {cout<< " Ac ";}
    virtual const A* j() {cout<< " A::j "; return this;}
    virtual void k() {cout<< " A::k "; m();}
    void m() {cout<< " A::m "; j();}
};

class C: virtual public B {
public:
    C() {cout<< " C() ";}
    ~C() {cout<< " ~C ";}
    void g() const {cout<< " C::g ";}
    void k() override {cout<< " C::k "; B::n();}
    virtual void m() {cout<< " C::m "; g(); j();}
    B& n() override {cout<< " C::n "; return *this;}
};

class E: public C, public D {
public:
    E() {cout<< " E() ";}
    ~E() {cout<< " ~E ";}
    E(const E& x) {cout<< " Ec ";}
    virtual void g() const {cout<< " E::g ";}
    const E* j() {cout<< " E::j "; return this;}
    void m() {cout<< " E::m "; g(); j();}
    D& n() final {cout<< " E::n "; return *this;}
};

A* p1 = new E(); B* p2 = new C(); A* p3 = new D(); B* p4 = new E();
const A* p5 = new D(); const B* p6 = new E(); const E* p7 = new E();
    
```

(dynamic\_cast<E\*>(p5)) -> j();  
 const E\*



## Esercizio Cosa Stampa

```
class A {
public:
    A() {cout<< " A() ";}
    ~A() {cout<< " ~A ";}
    A(const A& x) {cout<< " Ac ";}
    virtual const A* j() {cout<< " A::j "; return this;}
    virtual void k() {cout<< " A::k "; m();}
    void m() {cout<< " A::m "; j();}
};
```

```
class C: virtual public B {
public:
    C() {cout<< " C() ";}
    ~C() {cout<< " ~C ";}
    void g() const {cout<< " C::g ";}
    void k() override {cout<< " C::k "; B::n();}
    virtual void m() {cout<< " C::m "; g(); j();}
    B& n() override {cout<< " C::n "; return *this;}
};
```

```
class E: public C, public D {
public:
    E() {cout<< " E() ";}
    ~E() {cout<< " ~E ";}
    E(const E& x) {cout<< " Ec ";}
    virtual void g() const {cout<< " E::g ";}
    const E* j() {cout<< " E::j "; return this;}
    void m() {cout<< " E::m "; g(); j();}
    D& n() final {cout<< " E::n "; return *this;}
};
```

```
A* p1 = new E(); B* p2 = new C(); A* p3 = new D(); B* p4 = new E();
const A* p5 = new D(); const B* p6 = new E(); const E* p7 = new E();
```

```
class B: virtual public A {
public:
    B() {cout<< " B() ";}
    virtual ~B() {cout<< " ~B ";}
    virtual void g() const {cout<< " B::g ";}
    virtual const B* j() {cout<< " B::j "; n(); return this;}
    void k() {cout<< " B::k "; j(); m();}
    void m() {cout<< " B::m "; g(); j();}
    virtual A& n() {cout<< " B::n "; return *this;}
};
```

```
class D: virtual public B {
public:
    D() {cout<< " D() ";}
    ~D() {cout<< " ~D ";}
    virtual void g() {cout<< " D::g ";}
    const B* j() {cout<< " D::j "; return this;}
    void k() const {cout<< " D::k "; k();}
    void m() {cout<< " D::m "; g(); j();}
};
```

NON COMPILA → MANCA K();

(dynamic\_cast<E\*>(const\_cast<B\*>(p6)))->k();

↓

A\* p6 = new B

B\* p6 = new B

## Esercizio Cosa Stampa

```
class A {
public:
    A() {cout<< " A() ";}
    ~A() {cout<< " ~A ";}
    A(const A& x) {cout<< " Ac ";}
    virtual const A* j() {cout<< " A::j "; return this;}
    virtual void k() {cout<< " A::k "; m();}
    void m() {cout<< " A::m "; j();}
};
```

```
class C: virtual public B {
public:
    C() {cout<< " C() ";}
    ~C() {cout<< " ~C ";}
    void g() const {cout<< " C::g ";}
    void k() override {cout<< " C::k "; B::n();}
    virtual void m() {cout<< " C::m "; g(); j();}
    B& n() override {cout<< " C::n "; return *this;}
};
```

```
class E: public C, public D {
public:
    E() {cout<< " E() ";}
    ~E() {cout<< " ~E ";}
    E(const E& x) {cout<< " Ec ";}
    virtual void g() const {cout<< " E::g ";}
    const E* j() {cout<< " E::j "; return this;}
    void m() {cout<< " E::m "; g(); j();}
    D& n() final {cout<< " E::n "; return *this;}
};
```

```
A* p1 = new E(); B* p2 = new C(); A* p3 = new D(); B* p4 = new E();
const A* p5 = new D(); const B* p6 = new E(); const E* p7 = new E();
```

```
class B: virtual public A {
public:
    B() {cout<< " B() ";}
    virtual ~B() {cout<< " ~B ";}
    virtual void g() const {cout<< " B::g ";}
    virtual const B* j() {cout<< " B::j "; n(); return this;}
    void k() {cout<< " B::k "; j(); m();}
    void m() {cout<< " B::m "; g(); j();}
    virtual A& n() {cout<< " B::n "; return *this;}
};
```

```
class D: virtual public B {
public:
    D() {cout<< " D() ";}
    ~D() {cout<< " ~D ";}
    virtual void g() {cout<< " D::g ";}
    const B* j() {cout<< " D::j "; return this;}
    void k() const {cout<< " D::k "; k();}
    void m() {cout<< " D::m "; g(); j();}
};
```

new E(\*p7);

delete p1; ..

delete p4; ..

AC) B() C) D) Ec ←

~AC) ←

~B() ~D() ~C) ←

~B() ←