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

#### Esercizio Cosa Stampa

```
class B: virtual public A (
class A (
public:
                                                                                                 public:
  A() {cout<< " A() ";}
~A() {cout<< " ~A ";}
                                                                                                    B() {cout<< " B() ";}
                                                                                                    virtual "B() {cout<< " "B ";}
                                                                                                    virtual void g() const {cout << " B::g ";}
   A(const A& x) {cout<< " Ac ";}
                                                                                                   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;}</pre>
  virtual const A* j() {cout<<" A::j "; return this;}
virtual void k() {cout <<" A::k "; m();}</pre>
   void m() {cout <<" A::m "; j();}
class C: virtual public B {
                                                                                                 class D: virtual public B {
public:
                                                                                                 public:
  C() {cout<< " C() ";}
-C() {cout<< " -C ";}
                                                                                                    D() {cout<< " D() ";}
~D() {cout<< " ~D ";}
   void g() const {cout <<" C::g ";}</pre>
                                                                                                    virtual void g() {cout << " D::g ";}</pre>
  void k() override {cout <<" C::k "; B::n();}
virtual void m() {cout <<" C::m "; g(); j();}
B& n() override {cout <<" C::n "; return *this;}</pre>
                                                                                                    const B* j() {cout <<" D::j "; return this;}
void k() const {cout <<" D::k "; k();}</pre>
                                                                                                    void m() {cout <<" D::m "; g(); j();}</pre>
                                                                            NON COMPILA
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();}</pre>
                                                                                                   (dynamic_cast<const E*>(p1->j()))->g();
  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();
```

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 NESSIINA STAMPA

```
Esercizio Cosa Stampa
```

```
class A {
                                                                                                                                                               class B: virtual public A {
public:
                                                                                                                                                              public:
                                                                                                                                                                 ublic:
B() {cout<< " B() ";}
virtual "B() {cout<< " "B ";}
virtual void g() const {cout << " B::g ";}
virtual const B* j() {cout << " B::j ";}
void k() {cout << " B::k "; j() m(); }
void m() {cout << " B::m "; g(); j();}
virtual As n() {cout << " B::n "; return *this;}</pre>
    A() {cout<< " A() ";}
~A() {cout<< " ~A ";}
    "A() {cout<< " 'A ";}
A(const A6 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 {
                                                                                                                                                               class D: virtual public B {
                                                                                                                                                              public:
 public:
                                                                                                                                                                  ublic:
D() {cout<< " D() ";}
D() {cout<< " "D ";}
virtual void g() {cout <<" D::g ";}
const B+ j() {cout <<" D::j "; return this;}
void b() cout <<" D::j "; return this;}</pre>
    C() {cout<< " C() ";}
C() {cout<< " C";}
    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;}
                                                                                                                                                                    vo
                                                                                                                                                              1:
 class E: public C, public D {
public:
   E() {cout<< " E() ";}</pre>
      "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()
```

# Esercizio Cosa Stampa

```
class B: virtual public A (
 class A (
 public:
                                                                                            public:
                                                                                               ublic:
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;}</pre>
   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 D: virtual public B {
 class C: virtual public B {
public:
                                                                                            public:
    C() {cout<< " C() ";}
C() {cout<< " C";}
                                                                                               D() {cout<< " D() ";}
D() {cout<< " D ";}
   void g() const {cout <<" C::g ";}
void k() override {cout <<" C::k "; B::n();}</pre>
                                                                                                virtual void g() (cout <<" D::g ";)
                                                                                               const B* j() {cout << " D::j "; return this;}
void k() const {cout << " D::k "; k();}</pre>
    virtual void m() {cout <<" C::m "; g(); j();}
   B& n() override (cout <<" C::n "; return *this;)
                                                                                               void m() {cout <<" D::m "; g(); j();}</pre>
1:
 class E: public C, public D {
public:
   E() {cout<< " E() ";}
    "E() {cout<< " "E ";}
                                                                                    P2 -> 30->60>
   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();
```

# Esercizio Cosa Stampa

```
class B: virtual public A {
public:
                                                                                   public:
  A() {cout<< " A() ";}
~A() {cout<< " ~A ";}
                                                                                      B() {cout<< " B() ";}
virtual "B() {cout<< " "B ";}
                                                                                      virtual void g() const {cout <<" B::g ";}
virtual const B* j() {cout <<" B::j "; n(); return this;}</pre>
  A(const A& x) {cout<< " Ac ";}
  virtual const A* j() {cout<<" A::j "; return this;}
                                                                                      void k() {cout <<" B::k "; j(); m(); }
void m() {cout <<" B::m "; g(); j();}</pre>
  virtual void k() {cout <<" A::k "; m();}
  void m() {cout <<" A::m "; j();}
                                                                                      virtual A& n() {cout <<" B::n "; return *this;}
                                                                                    1:
                     (2)
class C: virtual public B {
                                                                                    class D: virtual public B {
public:
                                                                                   public:
  C() {cout<< " C() ";}
                                                                                      D() {cout<< " D() ";}
  "C() {cout<< " "C ";}
                                                                                      "D() {cout<< " "D ";}
                                                                                      virtual void g() {cout <<"D::g ";}
const B* j() {cout << D::j "} return this;}
void k() const {cout << D::k "; k();}</pre>
  void g() const {cout <<" C::g ";}</pre>
  void k() override {cout <<" C::k "; B::n();}
virtual void m() {cout <<" C::m "; g(); j();}</pre>
  B& n() override (cout <<" C::n "; return *this;)
                                                                                      void m() {cout <<" D::m "; g(); j();}
class E: public C, public D {
public:
  E() {cout<< " E() ";}
  "E() {cout<< " "E ";}
                                                                                                   P3 >K()
  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();}</pre>
  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();
```

#### Esercizio Cosa Stampa

```
class A (
                                                                                                                                 class B: virtual public A {
    A() {cout<< " A() ";}
                                                                                                                                    B() {cout<< " B() ";}
                                                                                                                                    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;}</pre>
   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 D: virtual public B public:
class C: virtual public B {
                                                                                                                                public:
   D() {cout<< " D() ";}</pre>
public:
    C() {cout<< " C() ";}
C() {cout<< " C";}
                                                                                                                                               {cout<< " "D ";}
   "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;}</pre>
                                                                                                                                    D() {cout<<" D::g ";}
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();}</pre>
class E: public C, public D {
public:
    E() {cout<< " E() ";}
-E() {cout<< " -E ";}
                                                                                                                                           P4 > N(), MD,
   "E() {cout<<" "E ";}

virtual void g() const {cout <<" E::g ";}

const E* j() {cout <<" E::j "; return this;}

void m() {cout <<" E:: "; g(); j();}
    D& n() final {cout <<" E::n "; return
                                                                                  *this: )
                                                                         را) د
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 > OUSTREAMS NON VA OW
```

# Esercizio Cosa Stampa

```
class A {
                                                                                                                  class B: virtual public A {
                                                                                                                  public:
public:
                                                                                                                     B() {cout<< " B() ";}
virtual "B() {cout<< " "B ";}
   A() {cout<< " A() ";}
~A() {cout<< " ~A ";}
                               A() ,;
   A() {cout<< " A ";}

A(const A6 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();}
                                                                                                                     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 A6 n() {cout << " B::n "; return *this;}</pre>
                                                                          &THIS/
                                                                                                                                                                                                                       B
 class C: virtual public B {
                                                                                                                  class D: virtual public B {
public:
                                                                                                                  public:
                                                                                                                                                                                                                       C() {cout<< " C() ";}
-C() {cout<< " -C ";}
                                                                                                                     D() {cout<< " D() ";}
D() {cout<< " D ";}
   "C() {cout<<" "C:;g ";

void g() const {cout <<" C:;g ";

void k() override {cout << (e::k'" B::n();}

virtual void m() {cout <<" C::m "; g(); j();}

B6 n() override {cout <<" C::n "; return *this;}
                                                                                                                     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();}</pre>
                                                                                                                                                                                                                                \nabla
                                                                                                                                                                                                                   C
class E: public C, public D {
                                                                                    3
   E() {cout<< " E() ";}
                                                                                                          ((dynamic_cast<D*>(p4))->n()).k();
   "E() {cout<< " "E ";}
E(const E& x) {cout<< " Ec ";}
D = PH = NOW 60) S::N() C::K
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();
```

# Esercizio Cosa Stampa

```
class A (
                                                                                                                                                                 class B: virtual public A {
                                                                                                                                                                     iblic:
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::m "; j(); m(); }
void m() {cout << " B::m "; g(); j();
virtual A& n() {cout << " B::m "; return *this;}</pre>
          ublic:
A() {cout<< " A() ";}
"A() {cout<< " A";}
A(const A6 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();}</pre>
       class C: virtual public B {
                                                                                                                                                                  class D: virtual public B {
      public:
   C() {cout<< " C() ";}
   TC() {cout<< " TC ";}</pre>
                                                                                                                                                                public:
   D() {cout<< " D() ";}
   TD() {cout<< " TD ";}</pre>
                                                                                                                                                                      D() [cout< - " ";)
virtual void g() [cout <<" D::g ";)
const 8* j() [cout <<" D::j "; return this;)
void k() const [cout <<" D::k "; k();)
void m() [cout <<" D::m "; g(); j();)
          void g() const {cout <<" C::g ";}
void k() override {cout <<" C::k "; B::n();}
virtual void m() {cout <<" C::m "; g(); j();}
Bé n() override {cout <<" C::n "; return *this;}</pre>
                                                                                                                                                                                                                              1 CONS 52
      class E: public C, public D {
      public:
                                                                                                                                       (dynamic_cast < £ *> (p5)) -> j();
          E() {cout<< " E() ";}
       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;}

D& n() final {cout << " E::n "; return *this;}
pl 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();
```

### Esercizio Cosa Stampa

```
class A (
                                                                    class B: virtual public A {
public:
                                                                     public:
  A() {cout<< " A() ";}
                                                                       B() {cout<< " B() ";}
  "A() {cout<< " "A ";}
                                                                       virtual B() {cout<< " B ";}
  A(const A& x) {cout<< " Ac ";}
                                                                       virtual void g() const {cout << " B::g ";}
                                                                       virtual const B* j() (cout <<" B::j "; n(); return this;)
  virtual const A* j() {cout<<" A::j "; return this;}
                                                                       void k() {cout <<" B::k "; j(); m(); }
void m() {cout <<" B::m "; g(); j();}</pre>
  virtual void k() {cout << " A::k "; m();}
  void m() {cout <<" A::m "; j();}
                                                                       virtual A& n() {cout << " B::n "; return *this;}
class C: virtual public B {
                                                                    class D: virtual public B {
public:
                                                                    public:
                                                                       D() {cout<< " D() ";}
  C() {cout<< " C() ";}
  "C() {cout<< " "C ";}
                                                                       D() {cout<< " "D ";}
                                                                       virtual void g() (cout <<" D::g ";)
  void g() const {cout <<" C::g ";}</pre>
                                                                       const B* j() {cout <<" D::j "; return this;}
void k() const {cout <<" D::k "; k();}</pre>
  void k() override {cout <<" C::k "; B::n();}</pre>
  virtual void m() {cout <<" C::m "; g(); j();}
  B& n() override (cout <<" C::n "; return *this;)
                                                                       void m() {cout <<" D::m "; g(); j();}
1:
                                                                NON COMPILA >> HANCA KL).
class E: public C, public D {
public:
  E() {cout<< " E() ";}
  "E() {cout<< " "E ";}
                                                          (dynamic_cast<E*>(const_cast<B*>(p6)))->k();
  E(const E& x) {cout<< " Ec ";}
  virtual void g() const {cout <<" E::g ";}
const E* j() {cout <<" E::j "; return this;}</pre>
  void m() {cout <<" E::m "; g(); j();}
  D& n() final {cout <<" E::n "; return *this;}
                                                                                             B 4 PG = N5W6
1;
                                                                                             5 + 96 znow 5 ()
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();
```

# Esercizio Cosa Stampa

```
class A {
                                                                                                                                  class B: virtual public A {
public:
                                                                                                                                  public:
                                                                                                                                      B() {cout<< " B() ";}
    A() {cout<< " A() ";}
                                                                                                                                   virtual "B() {cout<< " "B ";}
     "A() {cout<< " "A ";}
                                                                                                                                       virtual void g() const {cout << " B::g ";}
    A(const A& x) {cout<< " Ac ";}
    virtual const A* j() {cout<<" A::j "; return this;}
                                                                                                                                       virtual const B* j() (cout <<" B::j "; n(); return this;)
                                                                                                                                       void k() {cout <<" B::k "; j(); m(); }
void m() {cout <<" B::m "; g(); j();}</pre>
    virtual void k() {cout << " A::k "; m();}
    void m() {cout <<" A::m "; j();}
                                                                                                                                       virtual A& n() {cout << " B::n "; return *this;}
class C: virtual public B {
                                                                                                                                  class D: virtual public B {
public:
                                                                                                                                  public:
    C() {cout<< " C() ";}
                                                                                                                                      D() {cout<< " D() ";}
     C() {cout<< " "C ";)
                                                                                                                                       "D() {cout<< " "D ";}
                                                                                                                                      virtual void g() {cout <<" D::g ";}
    void g() const {cout <<" C::g ";}</pre>
                                                                                                                                      const B* j() {cout <<" D::j "; return this;}
void k() const {cout <<" D::k "; k();}</pre>
    void k() override {cout <<" C::k "; B::n();}</pre>
    virtual void m() {cout << " C::m "; g(); j();}</pre>
                                                                                                                                      void m() {cout <<" D::m "; g(); j();}
   B& n() override (cout << " C::n "; return *this;)
1:
class E: public C, public D {
public:
                                                                                   A() B() C() B() B() C() C()

(*p7)

(*p7)
    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();}
                                                                                                         ν5()νδ() ν(() ← | delete p4; ...
   D& n() final {cout << " E::n "; return *this;}
A* p1 = new E(); B* p2 = new C(); A* p3 = new D(); B* p4 = new E();
                                                                                                                                                   ~ 3C)
const A* p5 = new D(); const B* p6 = new E(); const E* p7 = new E();
```