

Esercizio 1

```

class A {
public:
    virtual ~A() {}
};

class B: public A {
public:
    virtual bool m() const =0;
};

class C: public B {
public:
    bool m() const {return true;}
};

class D: virtual public A {
protected:
    D() {}
};

class E: public D {
public:
    E& operator=(const E& x) {
        D::operator=(x);
        return *this;
    }
};
        
```

Handwritten Notes:

- ACD → IMPUCAZIONE
- C++ ≥ 17 → ACD = DEFAULT
- POLIMORFA
- ASSINATA
- CONCRETO
- NO COSTR. PUBBLICA
- CONCRETO
- C = E

Se c 'e un oggetto di tipo C allora in un main lo statement E e(c); compila correttamente

Si consideri il codice sopra. Selezionare quali delle seguente affermazioni sono **VERE**:

1. A è una classe base polimorfa A **V**
2. Il costruttore di copia di A non è pubblicamente disponibile **F**
3. A è una classe base astratta; **F**
4. Il costruttore senza parametri di A è polimorfo **F**
5. B è una classe derivata astratta **V**
6. B è una classe polimorfa **V**
7. L'assegnazione di B è virtuale pura **F**
8. B è una classe virtuale pura **F**
9. C è una sottoclasse concreta di B **V**
10. C è una classe polimorfa **V**
11. C è una classe che non permetta la costruzione pubblica dei suoi oggetti **F**
12. Nella classe C il metodo m() è overloaded **F**
13. D è una classe che non permette di costruire pubblicamente i suoi oggetti **V**
14. D è una classe che permette la costruzione di oggetti di D che siano sottooggetti di altri oggetti **V**
15. Se d è un oggetto di tipo D allora in un main() la chiamata d->m(); non compila **F**
16. D ha il costruttore di copia protetto **F**
17. E è una classe derivata indirettamente da D **F**
18. L'assegnazione ridefinita di E ha comportamento identico a quello dell'assegnazione standard di E **V**
19. Se c è un oggetto di tipo C allora in un main lo statement E e(c); compila correttamente **F**
20. E è una classe derivata indirettamente da D **F**
21. Se d è un oggetto di tipo D e c è un oggetto di tipo C allora in un main lo statement c=e; compila correttamente **V**

Handwritten Notes:

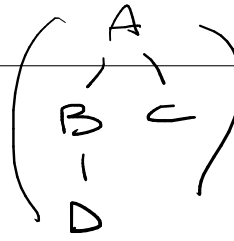
- OVERLOADING (SOVRACCARICO)
- VIRTUAL VOID BASE C() = 0
- VIRTUAL VOID BASE C()
- OVERLOADING (SOVRACCARICO)
- VIRTUAL METHOD C()
- VIRTUAL METHOD (PARAM)
- VIRTUAL METHOD (PARAM1, PARAM2)

Handwritten Note: BUILDOR

Esercizio 2

Siano A, B, C e D distinte classi polimorfe e si considerino le seguenti definizioni.

```
template<class A  
A funA(ref) { return &ref; }  
  
int main() {  
    B b;  
    fun<A>(b);  
    B* p = new D();  
    C c;  
    if (dynamic_cast<B*>(fun<A>(c))) cout << "bianco ";  
    else cout << "nero ";  
    if( !(dynamic_cast<D*>(new B())) ) cout << "rosso ";  
}
```



$A \not\leq B, C \not\leq B, C \leq A$
 $B \not\leq D$

Si supponga che:

1. il `main()` **compili correttamente ed esegua senza provocare errori a run-time;**
2. l'esecuzione del `main()` provochi in output su `cout` la stampa `nero rosso`.

Sotto queste ipotesi, selezionare per ognuna delle seguente relazioni di sottotipo $X \leq Y$ quali sono **sicuramente vere**:

- $A \leq B$
- $A \leq C$
- $A \leq D$
- $B \leq A$ V
- $B \leq C$
- $B \leq D$
- $C \leq A$ V
- $C \leq B$
- $C \leq D$
- $D \leq A$ V
- $D \leq B$ V
- $D \leq C$