

```
1: class Esame : protected vector<Studente*> {
2: public:
3:     Esame() {}
4:     void setTitolo(string t) { titolo = t;}
5:     ~Esame();
6:     Esame(const Esame& orig);
7:     Esame& operator=(const Esame& orig);
8:     void aggiungi(string m, string n, string c) { push_back(new Studente(m,n,c)); }
9:     bool rimuovi(Studente* s);
10:    unsigned int numStudenti() const;
11:    friend ostream& operator<<(ostream& out, const Esame& e);
12: private:
13:     string titolo;
14: };

//Completare la definizione della classe Studente (max 3 punti)
class Studente {
    public:

    private:
        string matricola;
        string nome;
        string cognome;
};

//Completare la definizione della classe StudenteLavoratore (max 3 punti)
class StudenteLavoratore : _____ Studente {
    public:

    private:

};

//Completare opportunamente il main (max 3 punti)
int main() {
    int esami=20;
    Esame* esame = new Esame[esami];
    for(int i = 0; i < esami; i++)
        esame[i].setTitolo("Esame"+i);

    return 0;
}
```

Programmazione Ad Oggetti. 29 Luglio 2016

Cognome: _____ Nome: _____
Matricola: _____ Esercizi lab: _____

Rispondere alle seguenti domande a risposta multipla (3 punti per risposta esatta, -2 punti per risposta sbagliata):

1. Quale tra le seguenti implementazioni è corretta?
 - a) `~Esame() {}`
 - b)

```
~Esame() {  
    for(int i=0; i < size(); i++)  
        delete (*this)[i];  
}
```
 - c) `~Esame() { delete *this; this->~end(); }`
 - d) `~Esame() { delete [] esame; }`
2. Quale tra le seguenti implementazioni è corretta?
 - a) `bool rimuovi(Studente* s){ bool res = this->remove(s); delete s; return res; }`
 - b) `void rimuovi(Studente* s){ pop_back(); delete s; return true; }`
 - c) `void rimuovi(Studente* s){ return s->remove(); }`
 - d)

```
bool rimuovi(Studente* s){  
    int j = 0; int i = 0;  
    for(; i < size(); i++) {  
        (*this)[j]=(*this)[i];  
        if(*at(i)!=*s)  
            j++;  
        else  
            delete (*this)[j];  
    }  
    resize(j);  
    return (i!=j);  
}
```
3. Quale metodo tra il costruttore di copia e l'operatore di assegnamento può essere rimosso?
 - a) Il costruttore di copia
 - b) L'operatore di assegnamento
 - c) Possono essere rimossi entrambi
 - d) Sono entrambi necessari
4. Quale tra le seguenti implementazioni è corretta?
 - a) `unsigned int numStudenti() const { return size(); }`
 - b)

```
unsigned int numStudenti() const {  
    int count = 0;  
    for(int i=0; i < capacity(); i++)  
        count++;  
    return ++count;  
}
```

- c) `unsigned int numStudenti() const { return capacity(); }`
d) `unsigned int numStudenti() const { return esame.size(); }`
5. Quale tra le seguenti implementazioni è corretta?
- a) `friend ostream& operator<<(ostream& out, const Esame& e) {
out << e << endl;
return out;
}`
- b) `friend ostream& operator<<(ostream& out, const Esame& e) {
for(int i = 0; i < e.size(); i++)
out << e[i] << endl;
return out;
}`
- c) `friend ostream& operator<<(ostream& out, const Esame& e) {
for(int i = 0; i < e.size(); i++)
out << *(e.operator[](i)) << endl;
return out;
}`
- d) `friend ostream& operator<<(ostream& out, const Esame& e) {
for(int i = 0; i < e.size(); i++)
out << e.esame[i] << endl;
return out;
}`
6. Sia `class EsameStraordinario : public Esame` la definizione di una classe `EsameStraordinario`. Quale tra le seguenti istruzioni non è consentita nel costruttore di `EsameStraordinario`?
- a) `push_back(NULL); (*this)[0]=new Studente("m","n","c");`
b) `cout << size() << endl;`
c) sono entrambe consentite
d) sono entrambe sbagliate
7. Quale tra le seguenti istruzioni sarebbe scorretta nella funzione `main`?
- a) `Esame e; e.push_back(NULL); e[0]=new Studente("m","n","c");`
b) `Esame e; cout << e->size() << endl;`
c) sono entrambe consentite
d) sono entrambe sbagliate