UNIVERSITÀ DELLA CALABRIA
DIPARTIMENTO DI MATEMATICA E INFORMATICA

Corso di Laurea In Informatica
Esame di Programmazione ad Oggetti
08/11/2018
Tempo a Disposizione: 60 min

Cognome: _____ Nome: _____

Matricola: _____ Esercizi lab (indicare quali e non quanti): _____

```
class Conferenza : private list<Partecipante*>
{
public:
 Conferenza (): numMaxRelatori(1), numRelatori(0)
                { relatori = new string[numMaxRelatori]; }
 Conferenza (const Conferenza &);
 ~Conferenza ();
 Conferenza & operator=(const Conferenza&);
 void add(Partecipante*); //aggiunge un dato partecipante
 void remove(Partecipante*); //rimuove un dato partecipante
 unsigned int size() const; //restituisce il numero di partecipanti
 void add(const string&); //aggiunge un dato relatore rispettando l'ordine alfabetico
                          //(è vietato l'uso di sort())
 void remove(const string&); //rimuove un dato relatore
 unsigned int getNumRelatori() const; //restituisce il numero di relatori
 friend bool operator==(const Conferenza& c1, const Conferenza& c2); //due conferenze sono uguali
                                          // se hanno gli stessi relatori
// Mostra per ogni partecipante non registrato il costo di registrazione che dipende dalla
tipologia di partecipante (ad esempio, studente, regolare, etc.)
void gestisciPartecipanti() {
     for(list<Partecipante*>::iterator it = begin(); it != end(); it++) {
          if(!(*it)->registrato())
              cout << (*it)->calcolaCostoRegistrazione() << endl;
     }
 }
private:
    string* relatori;
    unsigned int numMaxRelatori;
    unsigned int numRelatori;
};
//Fornire l'intestazione della classe Partecipante e della classe ParteciapanteStudente. (5 punti)
```

Dipartimento di Matematica e Informatica
Università della Calabria
Via P. Bucci, Cubo 30B
87036 Rende (CS) - ITALIA

Tel. (+39) 0984.496402
Fax (+39) 0984.496410
dipartimento.demacs@unical.it
dipartimento.demacs@pec.unical.it

www.mat.unical.it

Corso di Laurea In Informatica
Esame di Programmazione ad Oggetti
08/11/2018
Tempo a Disposizione: 60 min

```
//Implementare i seguenti metodi (max 13 punti)
 void Conferenza::remove(Partecipante* p){ //rimuove un dato partecipante









 }
 unsigned int Conferenza::size() const{ //restituisce il numero di partecipanti



 }
 void Conferenza::add(const string& relatore){
//aggiunge un relatore rispettando l'ordine alfabetico (è vietato l'uso di sort())
```

```
}
Conferenza& Conferenza::operator=(const Conferenza& c){
```

```
}
```

Dipartimento di Matematica e Informatica
Università della Calabria
Via P. Bucci, Cubo 30B
87036 Rende (CS) - ITALIA

Tel. (+39) 0984.496402
Fax (+39) 0984.496410
dipartimento.demacs@unical.it
dipartimento.demacs@pec.unical.it

www.mat.unical.it

Programmazione Ad Oggetti. 11 Novembre 2018

Cognome: _____ Nome: _____

Matricola: _____ Esercizi lab: _____

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

1. Quale/Quali tra le seguenti implementazioni è/sono corretta/corrette?

   a)
   ```
   Conferenza::Conferenza(const Conferenza& c) : numMaxRelatori(c.numMaxRelatori),
       numRelatori(c.numRelatori), list<Partecipante*>(c) {
       relatori = new string[numMaxRelatori];
       for(unsigned int i=0;i<numRelatori;++i)
           relatori[i] = c.relatori[i];
   }
   ```

   b)
   ```
   Conferenza::Conferenza(const Conferenza& c){
       numMaxRelatori=c.numMaxRelatori;
       numRelatori=c.numRelatori;
       relatori = new string[numMaxRelatori];
       for(unsigned int i=0;i<numRelatori;++i)
           relatori[i] = c.relatori[i];
   }
   ```

   c)
   ```
   Conferenza::Conferenza(const Conferenza& c){
       numMaxRelatori=c.numMaxRelatori;
       numRelatori=c.numRelatori;
       delete[] relatori;
       relatori=c.relatori;
       list<Partecipante*>=c.list;
   }
   ```

   d) `Nessuna delle precedenti.`

   Motivazione: _____

2. Quale/Quali tra le seguenti implementazioni è/sono corretta/corrette?

   a)
   ```
   void Conferenza::add(Partecipante* p){
       push_front(*p);
   }
   ```

   b)
   ```
   void Conferenza::add(Partecipante* p){
       insert(end(),*p);
   }
   ```

   c)
   ```
   void Conferenza::add(Partecipante* p){
       insert(begin(),p);
   }
   ```

   d)
   ```
   void Conferenza::add(Partecipante* p){
       push_back(p);
   }
   ```

   Motivazione: _____

3. Quale/Quali tra le seguenti implementazioni è/sono corretta/corrette?

a)
```
bool operator==(const Conferenza& c1, const Conferenza& c2){
    return c1==*this;
}
```

b)
```
bool operator==(const Conferenza& c1, const Conferenza& c2){
    if(c1.numRelatori!=c2.numRelatori)
        return false;
    for(unsigned int i=0;i<c1.numRelatori;i++)
        if(c1.relatori[i]==c2.relatori[i])
            return true;
    return false;
}
```

c)
```
bool operator==(const Conferenza& c1, const Conferenza& c2){
    if(c1.numMaxRelatori!=c2.numMaxRelatori)
        return false;
    for(unsigned int i=0;i<c1.numRelatori;++i)
        for(unsigned int j=0;i<c1.numRelatori;++j)
            if(i!=j && c1.relatori[j]!=c2.relatori[i])
                return false;
    return true;
}
```

d) Nessuna delle precedenti. L'implementazione corretta è:

Motivazione:

4. Quale/Quali metodo può/possono essere **const** nella classe Conferenza?

   a) `friend bool operator==(const Conferenza& c1, const Conferenza& c2);`

   b) `void add(Partecipante*);`

   c) `void remove(Partecipante*);`

   d) `Nessuno dei precedenti`

Motivazione: