UNIVERSITÀ DELLA CALABRIA
DIPARTIMENTO DI MATEMATICA E INFORMATICA

Corso di Laurea In Informatica
Esame di Programmazione ad Oggetti
22/11/2016
Tempo a Disposizione: 60 min

Cognome: _____ Nome: _____

Matricola: _____ Esercizi lab: _____

```
1: class Playlist{
 2: public:
 3:    Playlist():canzoni(0),numeroCanzoni(0),numeroMassimoCanzoni(0){};
 4:    Playlist(const Playlist& p);
 5:    ~Playlist();
 6:    void aggiungiCanzone(Canzone* c);
 7:    void riproduciProssimaCanzone();
 8:    Canzone* prossimaCanzone()  const;
 9:    unsigned int getNumCanzoniInCoda() const;
 10:   friend ostream& operator<<(ostream& out, const Playlist& p);
 11: protected:
 12:    Canzone** canzoni; //Coda di canzoni
 13:    unsigned int numeroCanzoni; //Numero di canzoni inserite
 14:    unsigned int numeroMassimoCanzoni; //Numero massimo di canzoni inseribili
};

//Completare la definizione della classe Canzone (max 2 punti)
class Canzone {
    public:









    private:



    protected:




};
//Completare opportunamente il main (max 2 punti)
int main() {
    Canzone* c1 = new Canzone("Ligabue","Urlando contro il cielo");
    Canzone* c2 = new Canzone("U2","Beautiful day");
    Playlist* p1 = new Playlist();
    Playlist* p2 = new Playlist();
    p1->aggiungiCanzone(c1);  p1->aggiungiCanzone(c2);
    p2->aggiungiCanzone(c2);









    return 0;
}
```

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
22/11/2016
Tempo a Disposizione: 60 min

```
//Implementare i seguenti metodi (max 5 punti)
Playlist::Playlist(const Playlist& p)
{









}
void aggiungiCanzone(Canzone* 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. 22 Novembre 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) `~Playlist() {canzoni.clear();}`

   b)
   ```
   ~Playlist() {
        for(int i = 0; i < numeroCanzoni; i++)
            delete canzoni[i];
   }
   ```

   c) `~Playlist() {delete canzoni;}`

   d) `Nessuna delle precedenti`

2. Quale tra le seguenti implementazioni è corretta?

   a)
   ```
   Canzone* prossimaCanzone() const {
        if(numeroCanzoni == 0)
             return 0;
        canzoni[0] = 0;
        return canzoni[0];
   }
   ```

   b)
   ```
   Canzone* prossimaCanzone() const {
        if(numeroCanzoni == 0)
             return 0;
        return canzoni[0];
   }
   ```

   c)
   ```
   Canzone* prossimaCanzone() const {
        return canzoni[numeroCanzoni];
   }
   ```

   d) `Nessuna delle precedenti`

3. Quale metodo dovrebbe necessariamente essere inserito nella classe Playlist?

   a) `L'operatore ==`

   b) `L'operatore =`

   c) `Sono entrambi necessari`

   d) `Sono entrambi inutili`

4. Quale tra le seguenti implementazioni è corretta?

   a) `unsigned int getNumCanzoniInCoda() const {return numeroCanzoni;}`

   b) `unsigned int getNumCanzoniInCoda() const {return numeroMassimoCanzoni;}`

   c) `unsigned int getNumCanzoniInCoda() const {return canzoni.size();}`

   d) `unsigned int getNumCanzoniInCoda() const {return size();}`

5. Quale metodo può essere virtuale puro nella classe Playlist?

   a) `void aggiungiCanzone(Canzone* c);`

   b) `void rimuoviProssimaCanzone();`

   c) `Entrambi i precedenti`

   d) `Nessuno dei precedenti`

6. Sia **class PlaylistRock: protected Playlist** la definizione di una classe PlaylistRock. Quale tra le seguenti istruzioni non è consentita nel costruttore di PlaylistRock?

   a) `cout << getNumCanzoniInCoda() << endl;`

   b) `canzoni = new Canzone*[10];`

   c) `Sono entrambe consentite`

   d) `Sono entrambe sbagliate`

7. Quale tra le seguenti istruzioni sarebbe scorretta nella funzione main?

   a)
   ```
   PlaylistRock* p = new Playlist();
   p->prossimaCanzone();
   ```

   b)
   ```
   PlaylistRock* p = new Playlist();
   cout << p->numeroCanzoni << endl;
   ```

   c) `Sono entrambe consentite`

   d) `Sono entrambe sbagliate`