

## **Programmazione**

PROVA SCRITTA – 4 Luglio 2018  
Parte I – LINGUAGGIO C++ e OOP

### **1 (8 punti)**

Si discuta la const correctness, con particolare riferimento ai puntatori e riferimenti del C++, oltre che ai metodi costanti.

### **2 (6 punti)**

Discutere dei membri statici di una classe. Come si dichiarano, definiscono e usano. Fare un esempio di una classe con attributo e metodo statico. Un metodo statico può accedere a un attributo non statico (motivare la risposta)?

### **3 (6 punti)**

Si descrivano le caratteristiche e differenze di overloading e overriding di metodi in C++. Quando si ha il caso di hiding ?

### **4 (10 punti)**

Descrivere gli smart pointer del C++, con particolare riferimento a unique e shared pointer. Come si usano ?

**Programmazione**PROVA SCRITTA – 4 Luglio 2018  
Parte II – PROGRAMMAZIONE**5 (12 punti)**

Si considerino le due classi base astratte `GameCharacter` e `Weapon` e se ne derivino le classi `JediKnight`, `RebelGuard` e `StormTrooper` (da `GameCharacter`), `Blaster` e `LaserSword` (da `Weapon`).

Si implementi un Abstract Factory, disegnandone il diagramma UML di classe (2 punti), per la creazione dei personaggi di un gioco in modo tale che solo i personaggi `JediKnight` abbiano una spada laser, mentre gli altri abbiano un `Blaster` (con strength 6 per `RebelGuard` e strength 8 per `StormTrooper`) come armi (8 punti). Scrivere un frammento di codice che crei un personaggio di ogni tipo (2 punti)

```
class GameCharacter {
public:
    virtual int fight(GameCharacter& enemy) = 0;
    virtual void move(int x, int y) = 0;
private:
    Weapon* w;
    int posX, posY;
};

class Weapon {
public:
    virtual int use(GameCharacter& enemy) = 0;
private:
    int strength;
    bool shield;
};
```

## 6 (10 punti)

Si consideri la struct Articolo che rappresenta un oggetto da acquistare rappresentato da un nome e quantità. Si scriva una classe che rappresenta una lista della spesa con i seguenti metodi: printAll() che stampa tutti gli elementi presenti, indicando nome e quantità, add() che aggiunge un articolo e nel caso sia già presente ne modifica la quantità secondo quanto indicato dall'elemento aggiunto; remove() che cancella un articolo dato il suo nome, getQuantity che riporta la quantità di un articolo sulla base del suo nome. (6 punti)

```
struct Articolo {  
public:  
    std::string nome;  
    int qty; // quantità  
};
```

- Nel caso si usi una `std::map<Key K, MappedType T>` si ricorda la presenza dei seguenti metodi:

`T& operator[] ( const Key& key );` ritorna un riferimento al valore mappato sulla chiave equivalente a quella passato come argomento, effettuando un inserimento se tale chiave non esiste.

`iterator find( const Key& key );` cerca un elemento con chiave equivalente a quella indicata; rende l'iteratore all'elemento se presente o all'iteratore prima posizione dopo la fine se non presente (`end()`).

`void erase( iterator pos );` rimuove l'elemento alla posizione indicata dall'iteratore

e che i valori presenti nella mappa sono del tipo: `std::pair<const Key, T>` con attributi pubblici `first` per la chiave e `second` per il valore associato.

- Nel caso si usi una `std::list<ValueType T>` si ricorda la presenza dei seguenti metodi:

`void push_back( const T& value );` aggiunge l'elemento in coda alla lista

`iterator erase( iterator pos );` rimuove l'elemento alla posizione indicata dall'iteratore

e l'algoritmo `find`:

```
template< class InputIt, class T >  
InputIt find( InputIt first, InputIt last, const T& value );
```

che cerca nell'intervallo tra l'iteratore `first` ed il `last` un valore. L'algoritmo usa l'operatore `==` che deve essere definito per il tipo su cui si applica.

Si ricorda che la firma di tale operatore è:

```
bool operator==(const T & right) const
```

## 7 (18 punti)

Scrivere una classe che rappresenti un file con nome, estensione e date di creazione, modifica ed ultimo accesso (6 punti). La data deve includere giorno, mese e anno + ora, minuto e secondo; creare la classe data (3 punti).

Implementare una classe directory con nome, data di creazione, modifica e ultimo accesso. Una directory deve contenere una collezione di file e di altre directory (6 punti)

Indicare se le tre classi hanno bisogno di implementare il costruttore di copia, l'operatore di assegnazione ed un distruttore e motivare la risposta. (3 punti)