Esercizi di Programmazione ad Oggetti

Lista n. 1

Esercizio 1

Definire una classe `IntMod` i cui oggetti rappresentano numeri interi modulo un dato intero n, che deve essere dichiarato come campo dati statico.
Definire metodi statici di `set_modulo()` e `get_modulo()` per tale campo dati statico.
Devono essere disponibili gli operatori di somma e moltiplicazione tra oggetti di `IntMod`.
Definire inoltre opportuni convertitori di tipo affiché questa classe sia liberamente usabile assieme al tipo primitivo `int` e valga la seguente condizione:
quando in una espressione compaiono interi e oggetti di `IntMod` il tipo dell'espressione dovrà essere intero.

Scrivere infine un programma d'esempio che utilizza tutti i metodi della classe.

Esercizio 2

Il seguente programma compila. Quali stampe produce la sua esecuzione?

```
#include<iostream>
using std::cout;

class A {
private:
  int x;
public:
  A(int k = 5): x(k) {cout << k <<  " A01 ";}
  A(const A& a): x(a.x) {cout << "Ac ";}
  A g() const {return *this;}
};

class B {
private:
  A ar[2];
  static A a;
public:
  B() {ar[1] = A(7); cout << "B0 ";}
  B(const B& b) {cout << "Bc ";}
};
A B::a = A(9);

A Fun(A* p, const A& a, B b) {
  *p = a;
  a.g();
  return *p;
};

main() {
  cout << ``ZERO\n'';
  A a1; cout << "UNO\n";
  A a2(3); cout << "DUE\n";
  A* p = &a1; cout << "TRE\n";
  B b; cout << "QUATTRO\n";

  a1 = Fun(p,a2,b); cout << "CINQUE\n";

  A a3 = Fun(&a1,*p,b); cout << "SEI";
}
```

Esercizio 3

Perché il seguente programma non compila? Modificare o eliminare una e soltanto una delle righe 1-8 in modo che il programmi compili.

```
class C {
public:                          // 1
  int *const p;                  // 2
  C(int a=0): p(new int(a))      // 3
  { }                            // 4
};

main() {
  C x(3);                        // 5
  C y;                           // 6
  x=y;                           // 7
  C z(y);                        // 8
}
```

Esercizio 4

Il seguente programma compila ed esegue correttamente. Quale stampa di output provoca?

```
#include<iostream>
#include<string>
using std::string; using std::cout;

class C {
private:
  int d;
public:
  C(string s=""): d(s.size()) {}
  explicit C(int n): d(n) {}
  operator int() {return d;}
  C operator+(C x) {return C(d+x.d);}
};

main() {
  C a, b("pippo"), c(3);
  cout << a << ' ' << 1+b << ' ' << c+4 << ' ' << c+b;
}
```

Esercizio 5

Definire, separando interfaccia ed implementazione, una classe `Raz` i cui oggetti rappresentano un numero razionale $\frac{num}{den}$ (naturalmente, i numeri razionali hanno sempre un denominatore diverso da 0). La classe deve includere:

1. opportuni costruttori;
2. un metodo `Raz inverso()` con il seguente comportamento: se l'oggetto di invocazione rappresenta $\frac{n}{m}$ allora `inverso` ritorna un oggetto che rappresenta $\frac{m}{n}$;
3. un operatore esplicito di conversione al tipo primitivo `double`;
4. l'overloading come metodi interni degli operatori di somma e moltiplicazione;

5. l'overloading come metodo interno dell'operatore di incremento postfisso, che, naturalmente, dovrà incrementare di 1 il razionale di invocazione;
6. l'overloading dell'operatore di output su `ostream`;
7. un metodo statico `Raz uno()` che ritorna il razionale 1.

Definire un esempio di `main()` che usi tutti i metodi della classe.

Esercizio 6

Il seguente programma compila ed esegue correttamente. Quali stampe provoca in output?

```
#include<iostream>
#include<string>
using std::string; using std::cout;

class B {
public:
  string s;
  B(char x='a', char y='b') {s += x; s += y; cout << "B012 ";}
  B(const B& obj): s(obj.s) {cout << "Bc "; }
};

class C {
private:
  B t;
  B* p;
  B u;
public:
  string s;
  C(char x='c', B y = B('d')): u(y), s(y.s) {
    s += x;
    cout << s[s.size()-2] << " C012 ";
  }
};

B F(B x, C& y) {
  (x.s) += (y.s)[0];
  return x;
}

main() {
  B b('e'); cout << "UNO\n";
  C c1('f',b); cout << "DUE\n";
  C c2; cout << "TRE\n";
  b=F(b,c2); cout <<"QUATTRO\n";
  cout << b.s  << " CINQUE";
}
```

Esercizio 7

Si consideri il seguente programma.

```
#include<iostream>
using std::cout;

class C {
```

```
public:
  int x;
  C(int k=5): x(k) {};
  C* m(C& c) {
    if((c.x != 5) && (x==5)) return &c;
    return this;
  }
};

main() {
  C a, b(2), c(a);
  cout << (b.m(b))->x << ' ' << (a.m(a))->x << ' ' << (b.m(c))->x
       << ' ' << c.m(a) << ' ' << c.m(c);
}
```

Il seguente programma compila correttamente? Se sì, al sua esecuzione quali stampe provoca in output?

Esercizio 8

Definire, separando interfaccia ed implementazione, una classe `Data` i cui oggetti rappresentano una data con giorno della settimana (lun-mar-...-dom). La classe deve includere:

- opportuni costruttori
- metodi di selezione per ottenere giorno della settimana, giorno, mese, anno di una data
- l'overloading dell'operatore di output esternamente alla classe
- l'overloading dell'operatore di uguaglianza
- l'overloading dell'operatore relazionale < che ignori il giorno della settimana
- un metodo `aggiungi_uno()` che avanza di un giorno la data di invocazione. Esempi: `lun 21/10/2002 => mar 22/10/2002`; `gio 31/1/2002 => ven 1/2/2002`; `mar 31/12/2002 => mer 1/1/2003`. Ignorare gli anni bisestili

Esemplificare l'uso della classe e di tutti i suoi metodi tramite un esempio di `main()`.