

## Esercizi di Programmazione 2

a.a. 06/07, 12/10/2006

### 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 affinché 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 programma compili.

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

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);
}
```

1. stampa in output 2 2 5 e di seguito 2 indirizzi diversi
2. stampa in output 2 2 5 e di seguito 2 indirizzi uguali
3. stampa in output 2 5 2 e di seguito 2 indirizzi uguali
4. non compila

### Esercizio 6

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()`.