Schriftelijk Examen:
"Structuur van Computerprogramma's II"

Eerste zittijd, academiejaar 2012-2013
Coen De Roover, Vrije Universiteit Brussel

18 januari 2013

Het examen bestaat uit 5 vragen, verdeeld over 4 bladzijden. Lees alle vragen aandachtig. Zet je naam, studierichting en rolnummer op **elke** pagina die je afgeeft. Veel succes!

Vraag 1: Geheugenbeheer Geef een voorbeeld van situaties waarin het gebruik van volgende concepten aangewezen is. Illustreer aan de hand van C++ code.

a) static local variable

b) user-defined pointer member access operator (bijvoorbeeld `class X { Y* operator->() { ...} };` waarbij `X` en `Y` user-defined types zijn)

Vraag 2: User-defined Types Bespreek bondig het verband tussen volgende concepten. Illustreer aan de hand van C++ code.

a) member initialization list

b) assignment operator

Vraag 3: Objectgericht programmeren Bespreek bondig volgende termen aan de hand van C++ code:

a) pure virtual member function

b) virtual destructor

Vraag 4: Generisch programmeren Vervolledig volgend programma met

a) een template function `copy_if` van 4 parameters die elk element (in de iterator range aangegeven door de eerste twee parameters) waarvoor een oproep van de laatste parameter slaagt, kopieert naar de iterator range beginnende op de derde parameter

b) een template class `NotYetEncountered` waarvan elke instantie bij toepassing op een waarde teruggeeft of zij al op deze waarde toegepast werd

```
#include <iostream>
#include <vector>
using namespace std;

// te vervolledigen met definities voor a) en b)

template <typename T>
inline bool alwaystrue(const T& x) { return true; }

int main() {
  vector<int> v(4);
  v[0] = 1;
  v[1] = 2;
  v[2] = 1;
  v[3] = 3;

  //copy all elements to v1
  vector<int> v1;
  copy_if(v.begin(),v.end(),
          back_insert_iterator<vector<int> >(v1),alwaystrue<int>);
  cout << v1.size() << endl; //prints 4

  //copy unique elements to v2
  vector<int> v2;
  copy_if(v.begin(),v.end(),
          back_insert_iterator<vector<int> >(v2),NotYetEncountered<
              int>());
  cout << v2.size() << endl;  //prints 3
  return 0;
}
```

Merk op dat `copy_if` als vierde parameter zowel een instantie van je template class als een functie moet kunnen aanvaarden.

Vraag 5: Programmabegrip Voorspel de uitvoer van de volgende programma's:

a) Betreffende controlestructuren:

```
#include <iostream>
using namespace std;

int main() {
  for (int j = 5; j > 0; j--) {
    int i(0);
    if(j % 2 == 0) continue;
    while(++i<2)
      if (i > 0) break;
    cout << "j=" << j << "i=" << i << endl;
  }
  return 0;
}
```

b) Betreffende arrays en pointers:

```
#include <iostream>
using namespace std;

void f(int* p, int* q, int** r) {
  *p = *q;
  *r = q++;
}

int main() {
  int a(5);
  int b[] = {2, 3};
  int* c;
  f(&a, b, &c);
  cout << "a=" << a << "*c=" << *c << endl;
  return 0;
}
```

c) Betreffende objectgericht programmeren:

```
#include <iostream>
using namespace std;

struct Feed {
  virtual int level() { return 1; }
};

struct UrlFeed : public Feed {
  virtual int level()  { return 2; }
};

int main() {
  Feed* f1 = new UrlFeed();
  Feed f2 = *f1;
  cout << f1->level() << "," << f2.level() << endl;
  return 0;
}
```

d) Betreffende object lifetime:

```
#include <iostream>
using namespace std;

struct A {
  A() { cout << "cA" << endl; }
  ~A() { cout << "dA" << endl; }
};

struct B : A {
  B() { cout << "cB" << endl; }
  ~B() { cout << "dB" << endl; }
};

int main() {
  B b;
  return 0;
}
```