Schriftelijk Examen:
"Structuur van Computerprogramma's II"

Eerste zittijd eerste semester, academiejaar 2011-2012

Coen De Roover, Vrije Universiteit Brussel

20 januari 2011

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

Vraag 1: Geheugenbeheer Bespreek bondig volgende termen aan de hand van C++ code:

a) memory leak

b) dangling pointer

Vraag 2: Functies Bespreek bondig volgende termen aan de hand van C++ code:

a) function overloading

b) function overriding

Vraag 3: Objectgericht programmeren Bespreek de omstandigheden waarin volgende vormen van overerving aangewezen zijn:

a) virtual inheritance

b) private inheritance

Vraag 4: Generisch programmeren Vervolledig volgend programma met

a) een template function `foreach`

b) een template class `Printer`

zodanig dat zowel de elementen van de array `a` als de elementen van de vector `v` geprint worden op `cout`:

```
#include <iostream> //voor cout
#include <vector>

using namespace std;

//hier te vervolledigen met definities voor a) Printer en b) foreach

int main() {
  Printer<int> p;

  vector<int> v;
  v.push_back(10);
  v.push_back(20);
  foreach(v.begin(),v.end(),p);

  int a[] = {10, 20};
  foreach(a,a+2,p);
  return 0;
}
```

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

a) Betreffende controlestructuren:

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

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

b) Betreffende exception handling en virtual functions:

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

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

struct SeriousError : public Error {
  virtual int level()  { return 2; }
};

void g() { cout << "ga"; throw SeriousError(); cout << "gb"; }

void f() { cout << "fa"; g(); cout << "fb"; }

int main() {
  try {
    cout << "ma";
    try {
      f();
      cout << "mb";
    } //end of inner try
    catch(Error e) { cout << e.level(); throw; }
  } //end of outer try
  catch(SeriousError& e) { cout << e.level(); }
  catch(...) { cout << 42; }
  cout << endl;
  return 0;
}
```

c) Betreffende arrays en pointers:

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

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

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

d) Betreffende constructoren en destructoren:

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

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

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

int main() {
  A a;
  return 0;
}
```