BERGISCHE UNIVERSITÄT
GESAMTHOCHSCHULE WUPPERTAL
GAUSS-STRASSE 20
42097 WUPPERTAL
(Korrespondenzanschrift)
42119 WUPPERTAL
(Lieferanschrift)
TELEX 8 592 262 bughw
TELEFAX (0202) 439-2901
TELEFON (0202) 439-1

Fachbereich 7
MATHEMATIK

Prof. Dr. Hans-Jürgen Buhl
Praktische Informatik / Numerik

e-mail: Juergen.Buhl@math.uni-wuppertal.de

Informatik I

WS98/99 – Übungsblatt 11

Bearbeitungstermin: 4. Kalenderwoche

Aufgabe 1. *Inkrement-/Dekrementoperatoren*

Testen Sie das folgende Programm:

```
//////////////////////////////////
// Datei:   incdec.cc
// Version: 1.0
// Zweck:   Nebeneffekte
// Autor:   Hans-Juergen Buhl
// Datum:   14.01.1999
//////////////////////////////////

#include <iostream>

//using namespace std;

int main()
{
  int i(13);
  cout << i;
  cout << endl;
  cout << i++;
  cout << endl;
  cout << i;
  cout << endl;
  cout << --i;
  cout << endl;
  cout << i;
  cout << endl;
  cout << endl;

  return 0;
}
```

Beschreiben Sie genau Zeile für Zeile den jeweiligen Ausdruckswert und den eventuell auftretenden Nebeneffekt.

Versuchen Sie, die Wirkungsweise des Programms zum Ausdruck `cout << --i++;` zu komprimieren und testen Sie. Interpretieren Sie das Ergebnis.

Aufgabe 2. *Vektoren mit dynamischer Dimension*

Um erst zur Laufzeit die aktuelle Dimension von Vektoren festzulegen, muß man folgendermaßen vorgehen:

```
//////////////////////////////////
// Datei:   dynvec.cc
// Version: 1.0
// Zweck:   dynamisch dimensionierte Vektoren
// Autor:   Hans-Juergen Buhl
// Datum:   28.09.1998
//////////////////////////////////

#include <iostream>

//using namespace std;

int main()
{
  int n;
  double* v;  // Variable für Startadresse für dyn. erzeugte Vektoren

  cout << "Bitte geben Sie die Dimension (ganze Zahl größer 0) ein: ";
  cin >> n;
  if (n <= 0) throw "n ist keine ganze Zahl größer 0";

  v = new double[n];
  for (int i=0; i < n; i++)
    v[i] = 2.0 * i;

  for (int j=0; j < n; j++)
    cout << "v[" << j << "] == " << v[j] << endl;

  delete []v;

  return 0;
}
```

Testen Sie das Programm. Was hat die Warnung

`"dynvec.cc", line 18: Warning: The variable v has not yet been assigned a value.`

für eine Bedeutung?

Ändern Sie das Programm wie folgt ab:

```
...
int n;
double* v;  // Variable für Startadresse für dyn. erzeugte Vektoren

cout << v[0] << endl;

cout << "Bitte geben Sie die Dimension (ganze Zahl größer 0) ein: ";
 ...
```

Welche Fehlermeldung erscheint zur Laufzeit und was hat sie zu bedeuten?

Ändern Sie das Programm erneut ab:

```
...
int n;
double* v(0);  // Variable für Startadresse für dyn. erzeugte Vektoren

cout << "Bitte geben Sie die Dimension (ganze Zahl größer 0) ein: ";
...
```

Warum sollte eine solche Initialisierung durch den Wert 0 immer durchgeführt werden?

Aufgabe 3. *Vektoren mit dynamischer Dimension (Forts.)*

Die Lebenszeit von mittels `new` erzeugten dynamischen Objekten ist nicht an die bisher bekannte Block-Regel gebunden; dynamische Objekte existieren solange (und belegen solange kostbaren Hauptspeicherplatz), bis sie explizit mittels `delete` wieder freigegeben werden.

Was passiert in einer Version der folgenden Art:

```
...
v = new double[n];
for (i=0; i < n; i++)
  v[i] = 2.0 * i;

for (j=0; j < n; j++)
  cout << "v[" << j << "] == " << v[j] << endl;

v = new double[15];

for (i=0; i < 15; i++)
  v[i] = 2.0 * i;

for (j=0; j < 15; j++)
  cout << "v[" << j << "] == " << v[j] << endl;


delete []v;
...
```


Starten Sie dazu den Debugger (`debugger dynvec`), wählen Sie unter „Execution/Run Time Checking“ den Punkt „Access & Leaks“ an, und starten Sie das Programm mit dem „run“-Knopf:

Was hat die erscheinende Fehlermeldung

```
Memory Leak (mel):
Found leaked block of size 8 bytes at address 0x23a70
At time of allocation, the call stack was:
[1] operator new() at 0xef7088c0
[2] main() at line 23 in "dynvec.cc"

Leak Summary:
actual leaks:         1  total size:       8 bytes
possible leaks:       0  total size:       0 bytes
```

zu bedeuten?

Wie können Sie dieses Speicherleck vermeiden? Testen Sie!

Aufgabe 4. *Eingebaute einfache Datentypen*

Stellen Sie je eine kurze Liste wichtiger Kennzahlen zu den auf Ihrem System verwendeten Integer- und Gleitkomma-Zahlentypen zusammen, indem Sie die Dateien `limits.h`, `float.h`, `ieee.h` und `math.h` zu Rate ziehen.