



# Einführung in die Informatik und Programmierung (Informatik I)

WS2000/2001 – Übungsblatt 4

15. November 2000  
Bearbeitungstermin: 47. KW

**Aufgabe 1.** *Euklidische Norm eines Vektors, 6 Punkte*

Schreiben Sie ein Programm, das mittels einer for-Zählschleife und der Funktionen *double sqrt(double)* sowie *double pow(double, int)* (aus *cmath*) die euklidische Norm eines Vektors aus *double<sup>dim</sup>* berechnet und ausdrückt.

**Aufgabe 2.** *Eigene Funktionen, 4 Punkte*

Im folgenden Programm sehen Sie, wie Sie eigene Funktionen deklarieren können:

```
#include <iostream>
#include <cassert>

using namespace std;

double average(double v[], int dim)
{
    assert(dim > 0);
    double result(0);

    for (int i = 0; i < dim; i++)
        result += v[i];

    return result/dim;
};

int main ()
{
    const int dim(6);
    double numbers[dim] = { 1, 2, 3, 4, 5, 6 };
}
```

```

    cout << average(numbers, 6) << endl;

    return 0;
}

```

Was tut dieses Programm? Testen Sie mit obigem Vektor die Ergebnisse von *average(numbers, 6)*, *average(numbers, 3)* und *average(numbers, 0)*. Nach Meldungen der Art

```

...
Abort (core dumped)

```

können Sie mittels der Anweisung „`dbx prgname core`“ den Hauptspeicher auszug „core“ genauer untersuchen (Programmzeile des Abbruchs). Verlassen Sie `dbx` mittels `exit` wieder.

Was passiert bei Aufruf von *average(numbers, 7)*, *average(numbers, 100)* bzw. *average(numbers, 1000)*? Raten Sie zunächst und testen Sie dann! Bitte vermerken Sie entsprechende Warnungen in ihrer Fehler-Kladde.

**Aufgabe 3.** *Ein einfach(st)es Programm in Variationen, 2 Punkte*

Geben Sie das folgende Programm ein, übersetzen Sie es, und bringen Sie es zur Ausführung:

```

#include <iostream>

using namespace std;

int main()
{

    cout << "Willkommen zum Programmierkurs" << endl;

    return 0;
}

```

Verfahren Sie ebenso mit den folgenden Programmen:

```

#include <iostream>
#include <string>

using namespace std;

int main()
{
    const string Begruessung("Willkommen zum Programmierkurs");

    cout << Begruessung << endl;

    return 0;
}

```

und

```
#include <iostream>
#include <string>

using namespace std;

int main()
{
    string Begruessung("Willkommen zum Programmierkurs");

    cout << Begruessung << endl;

    Begruessung = "Auf Wiedersehen!";
    cout << Begruessung << endl;

    return 0;
}
```

Was unterscheidet diese Varianten? Welche Vor- und Nachteile sehen Sie? Führen Sie insbesondere für das letzte Beispiel einen *Schreibtischtest* durch, in dem Sie jeweils den Speicherbelegungsstatus nach Bearbeitung jeder einzelnen Zeile beschreiben!

**Aufgabe 4.** *Eine weitere Variation, 2 Punkte*

Geben Sie das folgende Programm ein, übersetzen Sie es, und bringen Sie es zur Ausführung:

```
////////////////////////////////////
// Datei:   hello.cc
// Version: 1.0
// Zweck:   Ausgabe einer Nachricht
// Autor:   HJB
// Datum:   17.09.1998
////////////////////////////////////

#include <iostream>
#include <string>

using namespace std;

class Nachricht {

    const string Text;

public:

    Nachricht(const string& t) : Text(t) { };

    void print() const { cout << Text; };
};
```

```

};

int main()
{
    Nachricht Begruessung("Willkommen zum Programmierkurs");

    Begruessung.print();
    cout << endl;

    return 0;
}

```

Diskutieren Sie auch diese Problemlösung. Welche Vorteile sehen Sie?  
Ergänzen sie die Klasse `Nachricht` um eine Methode „`long getTextLength()`“, die die Länge des Nachricht-Textes mittels der `string`-Methode „`size_t length()`“ bestimmt und als Methoden-Ergebniswert zurückliefert.

**Aufgabe 5.** *Pseudozufallszahlen, 2 Punkte*

Modifizieren Sie das Ihnen schon bekannte Programm

```

////////////////////////////////////
// Datei:   random.cc
// Version: 1.0
// Zweck:   Pseudo-Zufallszahlen
// Autor:   Hans-Juergen Buhl
// Datum:   17.09.1998
////////////////////////////////////

#include <iostream>
#include <cstdlib>

using namespace std;

int main()
{

    srand(912345);

    for (int i=0; i<=20; i++)
        cout << rand() << endl;

    cout << endl << "Pseudozufallszahlen von 0.." << RAND_MAX << endl;

    return 0;
}

```

so, daß es Pseudozufallszahlen im Bereich  $[0.0,1.0]$  liefert und testen Sie.

Wie könnte man den Zufallszahlengenerator so initialisieren, daß bei jedem Programmstart ein anderer Initialisierungswert statt der festen 912345 benutzt wird? (nur Idee, noch nicht programmieren)

**Aufgabe 6.** *Klassendesign, 4 Punkte*

Konzipieren Sie (in UML-Klassendiagrammen) drei Klassen, die Ihnen für eine Problemlösung im Bereich „Software für die Seminarraumbelegung eines Fachbereichs der BUGHW“ mindestens nötig erscheinen.