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

Einführung in die Informatik und Programmierung (Informatik I)

WS2000/2001 – Übungsblatt 2

31. Oktober 2000
Bearbeitungstermin: 45. KW

Aufgabe 1. *Erkunden eines „unbekannten" Programms, 4 Punkte*

a.) Geben Sie das folgende Programm mit Hilfe des Texteditors `xemacs` ein, speichern Sie es in eine Datei des Namens `x.cc` ab, und übersetzen Sie es mittels

```
make x
```

in eine ausführbare Version.

```
////////////////////////////////
// Datei:   x.cc
// Version: 1.0
// Zweck:   Ausgabe einer Nachricht
// Autor:   Hans-Juergen Buhl
// 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;
}
```

b.) Was tut dieses Programm? (Führen Sie es dazu mittels `x` aus und beschreiben Sie möglichst genau die Ausgabe des Programmlaufs.)

Aufgabe 2. *Forts., 4 Punkte*

a.) Geben Sie das folgende Programm mit Hilfe des Texteditors `xemacs` ein, speichern Sie es in eine Datei des Namens `random.cc` ab, und übersetzen Sie es mittels

```
make random
```

in eine ausführbare Version.

```
////////////////////////////////
// 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;
}
```

b.) Was tut dieses Programm? (Führen Sie es dazu mittels `random` aus und beschreiben Sie möglichst genau die Ausgabe des Programmlaufs.)

Aufgabe 3. *Fehlermeldungen des Compilers, 4 Punkte*
Sollten Sie versehentlich `int j=0;` statt `int i=0;` geschrieben haben, so erhalten Sie eine Fehlermeldungsliste:

```
"random.cc", line 19: Error: i is not defined.
"random.cc", line 19: Error: i is not defined.
"random.cc", line 20: Error: cout is not defined.
"random.cc", line 20: Error: endl is not defined.
"random.cc", line 22: Error: cout is not defined.
"random.cc", line 22: Error: endl is not defined.
"random.cc", line 22: Error: endl is not defined.
7 Error(s) detected.
*** Error code 7
make: Fatal error: Command failed for target ‘random’
```

Diese Fehlerbeschreibung trifft recht gut zu. Es gibt aber auch relativ schlechte Fehlermeldungen. Bauen Sie absichtlich durch kleine textuelle Modifikationen syntaktische Fehler in das Programm ein, und notieren Sie sich die Fehlermeldungstypen mit den jeweiligen Fehlerursachen in einer extra Kladde.
Ergänzen Sie während des ganzen Semesters nach und nach diese Übersicht.

Aufgabe 4. *Problemlösen: Klassendesign, 4 Punkte*
Konzipieren Sie in Form von UML-Klassendiagrammen Objekte `Datum, Student, Person` und `Kraftfahrzeug`.

Aufgabe 5. *Forts., 2 Punkte*
Ergänzen Sie die Klassen `Kunde, Bestellung, Adresse, Artikel` um Klassen `Lieferschein, Rechnung`. Kennzeichnen Sie - wenn nötig - Attribute, Assoziationen bzw. Aggregationen.