Eidgenössische Technische Hochschule Zürich
Ecole polytechnique fédérale de Zurich
Politecnico federale di Zurigo
Swiss Federal Institute of Technology Zurich

Institut für theoretische Informatik
Dr. B. Gärtner

19. Oktober 2004

Informatik für Mathematiker und Physiker WS 04/05

URL: http://www.ti.inf.ethz.ch/ew/courses/Info1_04/

Vorlesungstermin: Dienstag 13.15–15.00 Uhr. Ort: HG F1.
Dozent: Bernd Gärtner <gaertner@inf.ethz.ch>, IFW B45.1.
Chefassistent: Michael Hoffmann <hoffmann@inf.ethz.ch>, IFW B47.2.
Sekretariat: Franziska Hefti <hefti@inf.ethz.ch>, IFW B49.1, Tel. 632 73 92.

Übungstermine:

Gruppe	Termin	Raum	Übungsleiter	Email
A	Di 15–17	LFW E13	Jörg Derungs	<derungs@inf.ethz.ch>
B	Di 15–17	HG D5.3	Leo Ruest	<ruestle@inf.ethz.ch>
C	Di 15–17	HG E1.2	Gábor Szabó	<szabog@inf.ethz.ch>
D	Di 15–17	HG F26.1	Michael Gatto	<gattom@inf.ethz.ch>
E	Di 15–17	HG F26.3	Uche Mennel	<umennel@student.ethz.ch>
F	Di 15–17	HG F26.5	Thomas Bruderer	<bruthoma@student.ethz.ch>
G	Di 15–17	LEC C14	Sarah Gugl	<sarah_gugl@gmx.ch>
H	Di 15–17	ML D13	Matthias Schrag	<mschrag@student.ethz.ch>
I	Do 15–17	ETZ J91	Reto Ghioldi	<ghioldir@student.ethz.ch>
J	Fr 15–17	HG D3.1	Alexander Hall	<hall@tik.ee.ethz.ch>
K	Fr 15–17	ML J37.1	Markus Deluigi	<mdeluigi@student.ethz.ch>

Achtung: In der ersten Semesterwoche finden keine Übungen statt. Die ersten Übungen sind am Di, 26.10.2004. Die erste Übungsserie ist aber dennoch bis zum 25.10.2004, 16:00 Uhr, abzugeben. Die Übungsgruppe C wird in englischer Sprache gehalten.

Für die Programmieraufgaben stehen die Rechner im HG D13 (13 Linux-Workstations), HG E29 (34 Sun-Workstations) sowie HG E26.1 (36 Sun-Workstations) zur Verfügung. Wo vorhanden, können natürlich auch eigene Rechner benutzt werden. Wer mit dem Gedanken spielt, sich ein eigenes Notebook anzuschaffen, der sollte einen Blick auf die Neptun Angebote werfen (http://www.neptun.ethz.ch/).

Wir arbeiten mit einer Unix Umgebung. Unter MS Windows kann man eine solche Umgebung sehr einfach mittels *cygwin* emulieren. Auf der Vorlesungswebseite (s.o.) findet man eine Kurzanleitung zur Installation von cygwin.

Testatbedingung: Am Ende der Übungen wird ein Testat vergeben. Das Testat erhält, wer

1. **sowohl** auf den ersten sechs **als auch** auf den folgenden – wahrscheinlich ebenfalls sechs – Übungsserien **mindestens 60%** der möglichen Punkte erreicht **sowie**
2. **mindestens die Hälfte** der Schnellübungen sinnvoll bearbeitet.

Prüfung: Die Prüfung besteht aus einer zweistündigen Klausur. Bedingung für die Zulassung zur Prüfung ist das Übungstestat. Es sind folgende Hilfsmittel erlaubt: maximal zwei handgeschriebene DIN A4 Blätter (beidseitig beschrieben, also max. vier A4 Seiten).

Abgabe der Übungen: Bearbeitungszeit ist in der Regel eine Woche, bei Programmieraufgaben manchmal auch zwei Wochen. Theoretische Aufgaben werden in der Pause in der Vorlesung abgegeben. Programmieraufgaben werden per Email an den Übungsleiter bis Montag 16.00 Uhr geschickt. Hierbei sind ein paar Punkte zu beachten, die weiter unten aufgelistet sind. Schriftliche Abgaben sind immer mit dem Namen der Vorlesung, der Nummer der Übungsgruppe (A...K), der Nummer des bearbeiteten Übungsblattes und dem eigenen Namen zu beschriften. Genügend Platz zwischen den Aufgaben und ein Rand hilft uns bei der Korrektur. Bitte keinen Bleistift oder roten Stift verwenden und bei Fehlern lieber sauber durchstreichen anstelle von Radieren oder Tipp-Ex.

Schnellübungen: Auf einigen Übungsserien wird die erste Aufgabe aus einer Schnellübung bestehen. Diese Aufgaben sind in den ersten x (typischerweise $x = 20$) Minuten der Übungsstunde zu bearbeiten und

werden dann vom Übungsleiter eingesammelt. Es versteht sich, dass hierfür die Anwesenheit in der Übungsstunde erforderlich ist, eine nachträgliche Abgabe ist **nicht** möglich. Schnellübungen werden in der Regel eine Woche vorher angekündigt.

Programmieraufgaben: Wie bereits erwähnt, erfolgt die Abgabe der Programmieraufgaben per Email an den Übungsleiter. Um uns die Verwaltungsarbeit zu erleichtern, sollten dabei ein paar Spielregeln beachtet werden.

- Nur die Programmtexte (*sourcecode*) verschicken, **nicht** die daraus mit Hilfe des Compilers generierten ausführbaren Programme (*executables*).
- Für jede Übungsserie nur eine Mail, und zwar mit *Subject* `"Serie <Nummer>"`, wobei `<Nummer>` entsprechend durch die Nummer der bearbeiteten Serie ersetzt wird, z.B. `"Serie 1"` . Sollte doch eine weitere Mail notwendig werden, weil Euch z.B. Fehler aufgefallen sind, so stellt bitte im Subject das Wort `"UPDATE:"` voran.
- Die verschiedenen Programmtexte werden jeweils als *attachments* vom Typ `Text/Plain` an die Mail angehängt.
- In Emails und Programmtexten: Nie mehr als 80 Zeichen pro Zeile!
- Programmtexte müssen vom Compiler ohne Fehlermeldungen akzeptiert werden.
- Jeder Programmtext enthält ganz am Anfang in Form eines Kommentars folgende Angaben: Programmname, Autor, Übungsgruppe und bearbeitete Übungsserie sowie Aufgabe. Beispiel:

```
// Informatik - Serie 13 - Aufgabe 4e
// Programm: my_program.C
// Autor:    X. M. Plestudent (Gruppe D)
```

- In den Programmen sind nur die Sprachkonstrukte zu verwenden, die in der Vorlesung oder Übung besprochen wurden.
- Programmtexte sind für Menschen geschrieben, auch wenn sie gleichzeitig von Rechnern verstanden werden müssen. Achtet deshalb auch auf das optische Erscheinungsbild Eurer Programme! Insbesondere zählt hierzu konsequentes und konsistentes Einrücken (*indenting*) und die ebensolche Verwendung von Leerzeichen z.B. vor und nach Operatoren und Schlüsselwörtern (*spacing*). Nehmt Euch die Programme aus der Vorlesung als Beispiel, dann kann eigentlich nicht viel schief gehen. ;-)

Zuguter Letzt: Bei Fragen wendet Euch bitte an Euren Übungsleiter oder bei weitergehenden Problemen an den Chefassistenten Michael Hoffmann. Email ist dafür ein bequemer Weg. Wir werden uns bemühen, diese einmal am Tag zu bearbeiten und zu beantworten, was bei Abwesenheit aber auch einmal länger dauern kann.

Es bleibt uns noch, Euch einen guten Anfang und viel Spass zu wünschen!

Literatur: (Polybuchhandlung, Raum MM B97.1)

Andrew Koenig and Barbara E. Moo: *Accelerated C++*, Addison-Wesley, 2000.

Stanley B. Lippman: *C++ Primer*, 3. Auflage, Addison-Wesley, 1998.

Bjarne Stroustrup: *The C++ Programming Language*, 3. Auflage, Addison-Wesley, 1997.