

# Programmieren in C / C++

## Wintersemester 2007/08

Hubert Feyrer, Florian Lutz

---

### Übung 6: Code lesen, IOCCC, Style Guides

#### Übung 6.1: Kontrollstrukturen

- Betrachten Sie <http://www.feyrer.de/PGC/src/ascii.c>
- Was macht es?
- Betrachten Sie den Quellcode!
- Welches Verfahren wird in jeder Tabelle verwendet, um das auszugebende Zeichen zu ermitteln?
- Wie wird die Tabellenstruktur sichergestellt?
- Compilieren und starten Sie das Programm!
- Wie ist der Code für die dritte Tabelle ohne ?: Operator zu schreiben?

#### Übung 6.2: Operatoren

- Betrachten Sie <http://www.de.ioccc.org/1998/bas2.c>
- Übersetzen Sie das Programm und starten Sie es:  

```
echo a | bas2
echo bla | bas2
echo blub | bas2
date | bas2
bas2 </etc/group
```
- Was macht das Programm?
- Welche Operatoren werden benutzt?
- Welche Bedeutung hat der & Operator?
- Aus wievielen (mit “;” abgeschlossenen) C-Statements besteht das Programm?
- Welche Funktionen werden aufgerufen?

- Welche Klammern müssen gesetzt werden, um die Präzedenz der einzelnen Operatoren explizit anzuzeigen?
- Welchen Fehler hat das Programm?
- Beheben Sie diesen!

### Übung 6.3: Berechnung

- Betrachten Sie <http://www.de.ioccc.org/2001/cheong.c>
- Übersetzen Sie das Programm!
- Starten Sie das Programm mit diversen Zahlen mit gerader Stellenzahl (ggf. führende Nullen benutzen):
 

```
cheong 01
cheong 02
...
cheong 09
cheong 10
cheong 11
...
cheong 99
cheong 0100
...
```
- Was macht das Programm?
- Betrachten Sie den Quellcode und analysieren Sie ihn!
- Welche Bedeutung hat “o”?
- Hinweise: <http://www.de.ioccc.org/2001/cheong.hint>

### Übung 6.4: Programmierstil

- Nehmen Sie sich Zeit, die Stil-Empfehlungen auf <http://pantransit.reptiles.org/prog/pikestyle.html> und <http://www.chris-lott.org/resources/cstyle/indhill-cstyle.html> durchzulesen!
- Vergleichen Sie diese Empfehlungen mit den beiden vorangegangenen Programmen (s.o., “Operatoren” und “Berechnung”) und weiteren IOCCC-Programmen (siehe <http://www.de.ioccc.org/years.html>)!

- Vergleichen Sie die Empfehlungen gegen
  - Linus Torvalds’s Linux Kernel Coding Style:  
(<http://pantransit.reptiles.org/prog/CodingStyle.html>)
  - Das NetBSD Style Guide:  
(<http://cvsweb.netbsd.org/bsdweb.cgi/~checkout~/src/share/misc/style>)
  - Den Mozilla Coding Style Guide:  
(<http://www.mozilla.org/hacking/mozilla-style-guide.html>)
- Unter welchen Bedingungen ist “guter” Programmierstil wichtig? Warum?