Hinweise zur Fehlersuche in C++

- Compiler meldet Syntaxfehler, jedoch nur solche nach Ausführung des Präprozessorlaufs.
- *Regel 1:* Nur den ersten Syntaxfehler beheben.
  Dann neu übersetzen.
- Vergessene Headerdateien führen unter Umständen zu langen Fehlermeldungen.
  Regel 2: Headerdateien nicht vergessen, vor allem `iostream` und `iomanip` nicht!
- *Regel 3:* Namespace-Anw. `using namespace std;` nicht vergessen!
- *Regel 4:* Systematisches Einrücken erleichtert die Überprüfung, ob die Blockstrukturen im Programm korrekt angelegt sind.
- *Regel 5:* Bei schwer lokalisierbaren Syntaxfehlern schrittweise Code auskommentieren.
  Kann auch bei Programmabstürzen hilfreich sein.

Hinweise zur Fehlersuche in C++ - Fortsetzung

- *Regel 6:* Bei falschen Ergebnissen Stelle suchen, ab der diese auftreten.
  Zu diesem Zweck zusätzliche Ausgabeanweisungen einfügen oder im Debugger Breakpoints definieren.
  Insbesondere Eingabedaten zur Kontrolle sofort wieder ausgeben.
- Liefert ein Programm bei gleichen Daten bei mehrfachem Aufruf unterschiedliche Ergebnisse, so sind wahrscheinlich nicht alle Variablen initialisiert.
  Regel 7: Überprüfen, ob alle Variablen (richtig) initialisiert sind.
  Entsprechende Warnungen des Compilers beachten und ggf. geeignete Optionen einschalten. (c++: `-O -Wall`).

Debuggerbenutzung bei Programmabstürzen

Meistens lässt sich die Stelle, an der ein Programm abstürzt, mit einem Debugger am einfachsten finden.

1. Damit das übersetzte Programm Informationen über Variablen- und Funktions*namen* enthält, sind spezielle Compileroptionen (z.B. c++: `-g`) erforderlich.
   Bsp.: `c++ -g prog.cpp`
2. Debugger starten (hier: gdb).
   Bsp.: `gdb a.out`
3. Programm (innerhalb des Debuggers) starten.
   Bsp.: `run`
   Erforderliche Eingaben vornehmen.
4. Aufrufverschachtelung im Abbruchzeitpunkt untersuchen.
   Bsp.: `where`
5. Debugger beenden.
   Bsp.: `quit`