The art of C++
Wie schreibe ich ordentlichen C++ Code?

- Konstanten : Benutze `const` wann immer möglich! Durch die Verwendung von `const` kann der Compiler effizienteren Code produzieren. Des Weiteren werden Fehler direkt beim Übersetzen erkannt, wenn versucht wird Variablen zu ändern die konstant sein müssen.
- Warnungen : Kompiliere deinen Code mit '-Wall -Wextra -Wshadow -Wconversion' und beseitige stets alle angezeigten Warnungen! Dies vermeidet Fehler, die zwar syntaktisch korrekt sind, aber sinnlos oder anders gemeint waren.
- Namensgebung : Vergebe passende Namen an alle Variablen und Funktionen! Es geht um die Verständlichkeit für andere Programmierer, die deinen Code wiederverwenden möchten. Lieber einen längeren Namen vergeben, der dann auch verständlicher ist, als einen zu kurzen Namen.
- Modularität : Fasse Funktionen in Header-Dateien zusammen, die etwas mit einander zutun haben und schreibe Source-Dateien dazu! Es ist ebenfalls sinnvoll darauf zu Achten nur Header-Dateien in andere Header- oder Source-Dateien einzubinden, die wirklich benutzt werden! Das erhöht die Lesbarkeit, Übersetzungsdauer und Wiederverwendbarkeit.
- Makros : Verwende keine Makros für Funktionen oder Variablen! Der Grund dafür ist, dass Makros als Textersetzung fungieren und eine Fehlerprüfung durch den Compiler nicht erfolgt, was zu falschen Code führt.
- Dokumentation : Dokumentiere deinen Code direkt beim schreiben! Dadurch verbessert sich die Lesbarkeit und das Verständnis des Codes für andere Programmierer und in den meisten Fällen für einen selbst. Es gibt Programme (z.B. doxygen) zum Verwalten der Dokumentation.
- Wiederverwendung : Bevor du neuen Code schreibst, vergewissere dich, dass dieser Code nicht bereits existiert! Quellen für bereits existierenden Code sind die STL (Standard Template Library), BLAS (Basic Linear Algebra Subroutines) oder Funktionen aus der Bibliothek an der gearbeitet wird.
- Speicherverwaltung : Produziere keine Speicherlöcher! Überprüfe deinen Code, ob stets aller mit `new` angelegter Speicher mit `delete` gelöscht wird. Es gibt Programme (z.B. valgrind) die solch eine Überprüfung durchführen.
- Konsistenz : Schreibe konsistenten Code! Dies gilt für die Namensgebung von Funktionen und Variablen sowie Parameterlisten, Absätze, Einrückung, Dokumentation, ... Führe vorhandenen Code konsistent fort.
- Editor : Verwende einen ordentlichen Editor (z.B. emacs)! Dies erleichtert die Programmierung.
- Makefile : Verwende Makefiles zur Kompilierung deines Codes! Die verringert den Aufwand beim kompilieren deutlich. Es gibt Programme (z. B. cmake) die Makefiles erstellen können.
- Testen : Teste deine Funktionen! Hierbei sollten neu geschriebene Funktionen unter bekannten Bedingungen getestet werden. Vor allem das Verhalten unter speziellen Parametern sollte getestet werden.