

# Labor Softwareentwurf mit Multiparadigmen-Programmiersprachen

## Versuch 1

### Versuch 1-3

- a)
- Prozedurale Programmierung:  
Aneinanderhaengen von Anweisungen, die dann Prozeduren ergeben.
  - Objektorientierte Programmierung:  
Gruppierung von Funktionen, Daten etc. in Objekte und zugehoerige Funktionen.
  - Generische Programmierung:  
Typen allgemein spezifizieren und erst bei Ausfuehrung festlegen, fuehrt zu mehr Wiederverwendbarkeit von Code.
  - Funktionale Programmierung:  
Anweisungen als mathematische Funktionen angeben.
- b) Ein *struct* ist ein zusammengesetzter Datentyp, welcher aus mehreren primitiven Datentypen besteht. *const* macht die Variable / Pointer zu seiner linken unveraenderbar.
- c) Der Praeprozessor in C/C++ ist ein Textersetzungsprogramm, um Makros/Direktiven in C/C++ Code umzuwandeln. Makros bergen das Problem, dass sie nicht durch die Syntax- Semantikanalyse des Programmcodes erfasst werden und die Fehlersuche schwieriger wird.
- d) Eine Referenz (&) gibt die Adresse einer Variable zurueck. Ein Pointer (\*) dereferenziert eine Variable und gibt den Inhalt der Referenz zurueck.
- e) Preprocessing(Include Dateien zusammensuchen), Compilation(Kompilieren was benoetigt wird), Linking(alles zusammenfuegen).

### Versuch 1-4

- b)
- **g++**: Aufruf des Compilers
  - **-std=c++11**: Auswaehlen des C Standards
  - **-Wall**: Alle Warnings
  - **-Wextra**: Zusaetzliche Warnings, die nicht in Wall enthalten sind.
  - **-o hello.out**: Benennung der Output-Datei
  - **hello-world.cc**: Angabe, was kompiliert werden soll.

### Versuch 1-7

Man muss `math.h` inkludieren.

### Versuch 1-8

- Probleme: Bei c), Angaben der Arraylaenge in der Funktionsdefinition, kann man laengere Arrays uebergeben.
- Probleme: Arrays koennen in C nur durch Call-by-Reference uebergeben werden. `sizeof` gibt daher nur die laenge des Pointers in byte aus. Daher kann man wie bei d) die Laenge des Arrays fest definieren, oder die Arraylaenge mit angeben.