

# Labor Softwareentwurf mit Multiparadigmen-Programmiersprachen

## Versuch 3

### Versuch 3-1

- a) Eine virtuelle Methode ist eine Methode, bei der erst zur Laufzeit bekannt wird welche Version benoetigt wird (Overriding!).
- b) Operationen und Funktionen die Teile einer class deklariert sind. Hierbei wird die Sichtbarkeit(scope) auf alle member einer class ausgeweitet. static member functions gehoeren nicht zu einem bestimmten object, und daher koennen sie auch nicht auf objektspezifische Variablen zugreifen.
- c) Der Destruktor bei C/C++ ist *delete*. Mittlerweile gibt es auch *unique\_ptr*, welche eine limitierte garbage-collection anbietet. Das Konstruktor / Destruktor-Pattern ist in Java nicht sinnvoll, da Java (mit Ausnahmen) die Speicherung und Loeschung von Programmdaten mit einem Garbage-Collector realisiert.
- d)
  - direct derivation: derive from class, can see protected from base-class, access modifier to set priv prot public access rights for base class methods, can only restrict, not increase access rights of base class functions and variables.
  - friend: declare method with that in public segment of class > able to access private variables & functions while not being a member method.
- e) Describing the proper declaring of variables or objects with const depending on their usage and safety.
- f) Ausnahmen sind ". (dot) :: ?: sizeof ". ? wurde aus logischen gruenden nicht ueberladen. Auf sizeof verlassen sich zu viele Funktionen. . / :: keine benefits, sondern nur verwirrung.

### Versuch 3-2

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

### Versuch 1-7

Man muss math.h inkludieren.

### Versuch 3-3

- a) Probleme: Bei c), Angaben der Arraylaenge in der Funktionsdefinition, kann man laengere Arrays uebergeben.
- b) 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.

### Versuch 3-4

- a) Probleme: Bei c), Angaben der Arraylaenge in der Funktionsdefinition, kann man laengere Arrays uebergeben.
- b) 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.

### **Versuch 3-5**

- a) Probleme: Bei c), Angaben der Arraylaenge in der Funktionsdefinition, kann man laengere Arrays uebergeben.
- b) 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.