

# C/C++ Materialpaket (Level AB)

## 11b\_PUTT – Putting it all together (Verständnisfragen)

Prof. Dr. Carsten Link

### Zusammenfassung

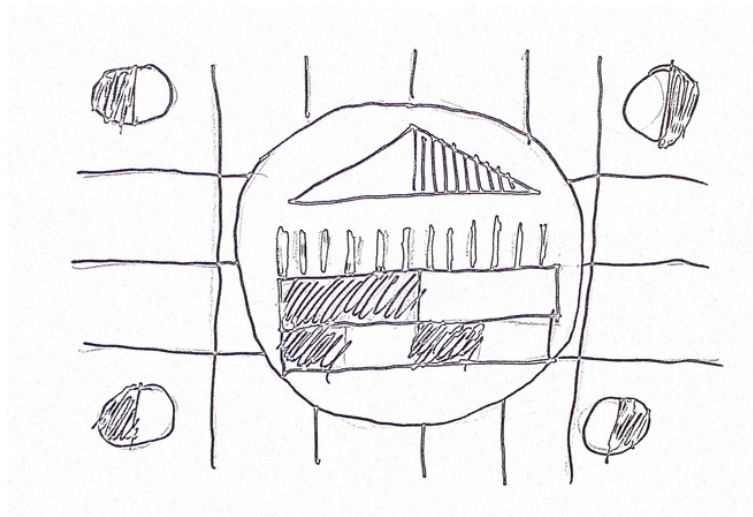


Abbildung 1: Testbild

## Inhaltsverzeichnis

|          |  |          |
|----------|--|----------|
| <b>1</b> | <b>Kompetenzen und Lernergebnisse</b>    | <b>2</b> |
| <b>2</b> | <b>Konzepte</b>                          | <b>2</b> |
| <b>3</b> | <b>Prüfungsvorbereitung</b>              | <b>2</b> |
| 3.1      | Verständnisfragen zu 01_ENV . . . . .    | 2        |
| 3.2      | Verständnisfragen zu 02_DATA . . . . .   | 3        |
| 3.3      | Verständnisfragen zu 03_FLOW_c . . . . . | 4        |
| 3.4      | Verständnisfragen zu 04_UDEF . . . . .   | 4        |
| 3.5      | Verständnisfragen zu 05c_CYCL . . . . .  | 4        |
| 3.6      | Verständnisfragen zu 06_BIND . . . . .   | 4        |

|                   |   |
|-------------------|---|
| 4 Nützliche Links | 4 |
| 5 Literatur       | 4 |

## 1 Kompetenzen und Lernergebnisse

Durch das Bearbeiten dieses Materialpaketes erwerben Sie diese Kompetenzen (Wissen, Fähigkeiten, Fertigkeiten zur Problemlösung):

**Sie können die isoliert vorgestellten Inhalte der vorangegangenen Materialpakete in Kombination anwenden.**

Die oben genannten Kompetenzen erwerben Sie, indem Sie Lernziele erreichen, welche sich prüfen lassen. Lernergebnisse: Sie können nachweislich<sup>1</sup>:

- Aufgaben, wie sie in einer Prüfung gestellt werden können und die alle vorangegangenen Themenbereiche betreffen, erfolgreich bearbeiten
- Fragen, wie sie in einer Prüfung gestellt werden können und die alle vorangegangenen Themenbereiche betreffen, richtig beantworten

## 2 Konzepte

Im Folgenden wird kein neuer Inhalt dargestellt. Vielmehr sollen Sie alles bereits Gelernte miteinander kombinieren, um les- und wartbare, effiziente C++-Programme entwickeln zu können.

## 3 Prüfungsvorbereitung

In diesem Materialpaket finden sich Aufgaben und Verständnisfragen, die über jene hinausgehen, die sich in den vorangegangenen Materialpaketen befinden, da sie Wissen und Fertigkeiten benötigen, die über das im jeweiligen Materialpaket Vorgestellte hinausgehen.

### 3.1 Verständnisfragen zu 01\_ENV

1. Sie haben zwei Dateien `main.cpp` und `func1.cpp` mit entsprechenden Header-Dateien. In `func1.cpp` ist eine Funktion `polynom1` definiert. Beim Erzeugen eines ausführbaren Programms bekommen Sie eine Fehlermeldung `unresolved external: polynom1` (oder `undefined symbol`). Was ist die Ursache?
2. Lückentext. Füllen Sie die Lücken in folgendem Text: Ein C++-Programm besteht aus ... -Dateien und ... -Dateien. Der ... fügt in erstere letztere ein und ersetzt ... . Vom ... wird ... -Code erzeugt, der vom ... in ... -Code

---

<sup>1</sup>Sie können das Erzielen der einzelnen Lernergebnisse beispielsweise bei einem Testat im Praktikum oder einer Aufgabe in der Modulprüfung nachweisen

übersetzt wird. Dabei entstehen ...-Dateien, die vom ... mit den ... kombiniert werden und zu einem ausführbaren Programm zusammengesetzt werden. Der gesamte Vorgang wird vom ... gesteuert.

3. Nehmen Sie folgendes an: Sie kopieren `poly1.c` zu `poly3.c` und lassen in `main()` zusätzlich zu `polynom1()` den Rückgabewert von `polynom3()` ausgeben. Wenn Sie bei letzterem in `poly3.c` den Parametertyp und Rückgabebetyp auf `double` umstellen und weiter nichts machen, stellen sich diese Fragen: a) was passiert schlimmstenfalls beim Übersetzen? b) was passiert schlimmstenfalls zur Laufzeit?
4. Beantworten Sie für jedes Tool der Toolchain die Frage “welches Problem löst es?”
5. Bei einem Projekt, welches aus vielen Quelldateien besteht, erhalten Sie einen oder mehrere Fehler der Art `duplicate symbol`. Welche Ursachen kann es dafür geben?
6. Müssen Header-Dateien einzeln compiliert werden?
7. In welchem Zusammenhang wäre es sinnvoll, Header-Dateien einzeln zu compilieren?

### 3.2 Verständnisfragen zu 02\_DATA

1. Gegeben sei die Zeichenkette `char * s = "0123"`. Wenn Sie das zugrundeliegende Bitmuster im Speicher uminterpretieren lassen (z. B. mit `int i = (*((int*)s));` bewusst das Bitmuster falsch interpretieren), welche Werte ergeben sich für die Typen `char` und `int`?
2. Eine Funktion verfügt über eine lokale Variable `local_int` (ist darin deklariert); das Programm, in dem sich die Funktion befindet, hat eine globale Variable `global_int`. Worin unterscheiden sich `local_int` und `global_int`?
3. Wie oft existiert `local_int` während des Programmlaufs?
4. Wie oft kann `local_int` während des Programmlaufs *gleichzeitig* existieren?
5. Stellen Sie sich eine Tabelle vor (z. B. aus dem Spiel “Stadt, Land, Fluss” oder eine Tabelle, welche nur Zahlenwerte enthält wie beispielsweise Umfrageergebnisse “Alter in Jahren, Gewicht in kg, Körpergröße in cm”). Ordnen Sie nun die beiden Aggregatstypen `struct` und `array` ein. Welche Möglichkeiten ergeben sich, die Tabelle im Speicher eines C++-Programmes darzustellen?
6. Welche Vor- und Nachteile hat es, Zahlen in Form von Zeichenketten zu speichern?
7. Welchen Typ haben die unten angegebenen Ausdrücke (bei mehreren: der letzte)?

```
1 char *s = "1.0"; *s
2 s[2]
3 4 / 3
```

9. Erstellen Sie für verschiedene Datentypen eine Liste von typischen Werten

(in der Art *explain like im five*)

### 3.3 Verständnisfragen zu 03\_FLOW\_c

1. Welche Vorteile hat es, einen Algorithmus rekursiv statt iterativ umzusetzen?

### 3.4 Verständnisfragen zu 04\_UDEF

1. Vergleichen Sie die Funktionen `printVehicleInfo()` und `printMinMax()`.
2. Was sind Unterschied und Gemeinsamkeit bei Copy Constructor (`X::X(const X&)`) und Zuweisungsoperator (`x1 = x2`)?

### 3.5 Verständnisfragen zu 05c\_CYCL

1. Wieviel Kopien des Typs `X` werden erzeugt beim Aufruf einer Funktion `X foo(X x)` erstellt? Lassen Sie außer Acht, dass der Compiler Kopien wegoptimieren kann (Compileroption `-fno-elide-constructors` und `-std=c++14`)
2. Welche Probleme können sich bei der automatischen Allokation von Objekten auf dem Stack auf Rechnern mit eingeschränkten Ressourcen ergeben?

### 3.6 Verständnisfragen zu 06\_BIND

1. Diskutieren Sie eine Umsetzung von `FizzBuzz` mittels Funktionszeigern.

## 4 Nützliche Links

- Stack Overflow: <http://stackoverflow.com>
- C++ reference: <http://en.cppreference.com/w/>
- C++ Referenz: <http://de.cppreference.com/w/>

## 5 Literatur

- [PPP] Stroustrup, Bjarne: Programming - Principles and Practice using C++
- [TCPL] Stroustrup, Bjarne: The C++ Programming Language, Fourth Edition
- [CUEB] U. Kirch, P. Prinz: C++ das Übungsbuch, Testfragen und Aufgaben mit Lösungen