

# 1. Übungsblatt zur Vorlesung „Einführung in die Programmiersprache C++“

## Ziele des Aufgabenblattes

In dieser Übung lernen Sie das Übersetzen und Linken von C-Programmen. Zudem machen Sie sich Gedanken über die Einbindung externen Bibliotheken in ein C-Projekt-

## Vorraussetzungen

### Unix-Umgebung:

Es wird erwartet, dass Sie sich Zugang zu einer Unix-Umgebung verschaffen. Dies können beispielsweise die Rechnerräume des Instituts für Informatik oder eine Linux bzw. Cygwin-Umgebung auf Ihrem privaten Computer sein. Weiterhin wird vorausgesetzt, dass Sie grundlegende Dateioperationen beherrschen. Mit dem Unix-Befehl `man` können Sie Hilfe zu alle C-Bibliotheksfunktionen bekommen.

### Entwicklungsumgebung:

Sie dürfen eine Entwicklungsumgebung Ihrer Wahl benutzen. Wie empfehlen Ihnen, Sich mit `eclipse` für C++ vertraut zu machen. Das CDT-Plugin zum Erstellen von C/C++-Projekten ist auf den Rechner im CIP-Pool installiert. Sollten Sie lieber einen Texteditor benutzen, empfehlen wir Ihnen sich mit `emacs` oder `vim` vertraut zu machen.

## Aufgabe 1.1: Hello1 (20 Punkte)

Schreiben Sie ein Programm mit dem Namen `hello1`, das den String „Hello World!“ auf dem Terminal ausgibt. Benutzen Sie dabei die `printf`-Funktion. Stellen Sie sicher, dass am Ende der Ausgabe ein Zeilenumbruch erfolgt. Übersetzen Sie Ihr Programm wie folgt:

```
gcc -Wall -Wstrict-prototypes -ansi -pedantic hello1.c -o hello1
```

Erklären Sie Ihrem Tutor die Bedeutung der Parameter.

## Aufgabe 1.2: make (10 Punkte)

Schreiben Sie ein Makefile zur Ihrem Programm aus Aufgabe 1.1. Ihr Makefile sollte neben dem Standard-Target (welches ist das?) auch die Targets `clean` (zum Entfernen des gerade erstellten Programms) und `test` (zum Ausführen des Programms) enthalten.

## Aufgabe 1.3: doxygen (10 Punkte)

Ergänzen Sie ihr Hello-World-Programm mit ausführlicher Inline-Dokumentation. Erzeugen Sie mit Hilfe von `doxygen` daraus eine Dokumentation als `html`-Seite und als `.pdf`. Machen Sie sich mit der Konfiguration von `doxygen` vertraut. Erklären Sie die Bedeutung der Tags in Ihrem Quelltext.

## Aufgabe 1.4: Einbindung externer Bibliotheken (60 Punkte)

In dieser Aufgabe machen Sie sich mit den grundlegenden Voraussetzungen zum Einbinden externer Bibliotheken vertraut. Beantworten Sie dazu die folgenden Fragen:

- Wo liegen unter Linux standardmäßig die vorhandenen Header- und Bibliotheksdateien?
- Welche Unix-Umgebungsvariable legt die Pfade fest, in denen gcc nach verfügbaren Header-Dateien sucht
- Wie heißt die entsprechende Variable für externe Bibliotheksdateien?
- Wie können diese Variablen sinnvoll gesetzt werden. Was müssen Sie tun, um diese dauerhaft zu setzen?
- Erklären Sie die Bedeutung der gcc-Parameter `-I`, `-L`, `-l`
- Was ist der Unterschied zwischen `#include "header.h"` und `#include <header.h>`?
- Erklären Sie den Unterschied zwischen statischen und dynamischen Linken. Wie können Sie statisches Linken erzwingen? Welche Konsequenzen hat das?
- Welche Informationen liefert Ihnen das Linux-Tool `ldd`?
- Testen Sie das Programm `nm` auf einer Bibliotheksdatei in Ihrem Linux-System. Erklären Sie die Ausgabe.
- Wie würden Sie vorgehen, wenn Sie für Ihr Programmierprojekt eine externe Bibliothek benötigen, aber keine Rechte haben, diese im System zu installieren?
- Für den Fall, dass Sie die benötigten Rechte haben, wieso sollten Sie es sich genau überlegen, ob Sie eine Bibliothek installieren sollten?
- Erklären Sie die generelle Bedeutung eines Paketmanagers.

## Abgabe

Drucken Sie Ihre Quelltexte und Antworten zu den Fragen aus und geben Sie diese bis Montag, den 20.10.2014, mit Angabe Ihres Namens und Ihrer Testatgruppe (A, B oder C) ab. Die Abgabe erfolgt individuell. Schicken Sie zusätzlich pro Arbeitsgruppe eine Mail mit den Unterlagen zum Testat an die Adresse [cpp@informatik.uni-osnabrueck.de](mailto:cpp@informatik.uni-osnabrueck.de). Geben Sie im Betreff ebenfalls Namen und Testatgruppe an.

