

Übungsblatt 1

Ralph Gauges

Ursula Rost

Katja Wegner

24.10.2007

Aufgabe 1:

Beim Schreiben von Programmen macht man selbstverständlich auch Fehler und seien es nur Tippfehler. Besonders Sprachen wie C++, welche sehr komplex sind, laden förmlich dazu ein Fehler zu machen. Da sich dies, insbesondere am Anfang, nicht verhindern lässt, ist es umso wichtiger, Fehler schnell zu finden und zu beseitigen. Die Fehlermeldungen des Präprozessors, des Compilers und des Linkers sind hierbei eine große Hilfe. Man sollte sich deshalb frühzeitig angewöhnen, genau auf solche Fehlermeldungen zu achten. Mit der Zeit lernt man dann welche Fehlermeldungen für welche Fehler typisch sind und kann diese schneller beseitigen.

Es ist ebenfalls immer ratsam, die Option `-Wall` beim kompilieren zu verwenden. Die `-Wall` Option veranlasst den Compiler zusätzliche Warnungen auszugeben, falls er auf Programmcode trifft, welcher zwar gültiges C++ darstellt, aber potentiell zu Problemen führen kann. Auch diese Warnungen sollte man ernst nehmen und die Ursachen nach Möglichkeit beseitigen.

Geben Sie folgende drei Versionen des "HelloWorld"Programms in den Texteditor Ihrer Wahl ein und übersetzen Sie sie. Jedes dieser Programme enthält einen Fehler und für jede Version des Programme unten kommt die Fehlermeldung von einem anderen Programm. Versuchen Sie herauszufinden welches Programm bzw. welche Programmkomponente (Präprozessor, Compiler, Linker) den Fehler meldet.

```
// HelloWorld 1
#include <iostream>

int main()
{
    std::cout << "Hello World!\n";
    return 0
}
```

```
// HelloWorld 2
#include <isostream>

int main()
{
    std::cout << "Hello World!\n";
    return 0;
}
```

```
// Hello World 3
#include <iostream>

int moin()
{
    std::cout << "Hello World!\n";
    return 0;
}
```

Aufgabe 2:

Schreiben Sie ein Programm mit zwei Variablen a und b vom Typ Integer. Initialisieren Sie a mit dem Wert 10 und b mit dem Wert 15. Tauschen Sie dann die Werte von a und b und geben das Ergebnis aus. Die Ausgabe des Programms sollte folgendermaßen aussehen:

```
a: 15  
b: 10
```

Aufgabe 3:

Schreiben Sie ein Programm, welches die Zahlen von 1-10 multipliziert und das Ergebnis ausgibt. Überprüfen Sie das Ergebnis mit einem Taschenrechner.

Aufgabe 4:

Schreiben Sie ein Programm, welches die Zahlen von 1-15 multipliziert und das Ergebnis ausgibt. Überprüfen Sie das Ergebnis mit einem Taschenrechner.

Aufgabe 5:

Schreiben Sie ein Programm, welches die Zahlen von 1-10 multipliziert und das Ergebnis ausgibt jedoch ohne dabei eine Form der Multiplikation (*, *=) zu benutzen.