Programmierkurs IPK SS23
Andreas Naumann
IWR, Universität Heidelberg

Übung 2
Abgabedatum —
ipk-exercises:2023-ss-1-ga451146

Übung 2

Aufgabe 1: Syntaxfehler und Compiler-Meldungen

In dieser Aufgabe arbeiten wir mit C++-Quellcode, aber Sie müssen diesen nicht wirklich verstehen. Es geht hier primär darum, die Fehlermeldungen des Compilers zu verstehen und die Tipps zum Beheben der Fehler richtig anzuwenden.

(a) Erstellen Sie die Datei `errors.cc` mit folgendem (absichtlich fehlerhaften) Inhalt:

```
#include <iostrea>

int main(int argc, char** argv)
  stf::cout << "Typing is difficult" << endl;
  int ret = 0;
  return retv
}
```

Versuchen Sie, das Programm zu kompilieren. Der Compiler wird diverse Fehlermeldungen ausgeben. Beheben Sie den oder die angezeigten Fehler und starten Sie den Compiler erneut. Dies kann unter Umständen dazu führen, dass der Compiler andere Fehler anzeigt. Machen Sie so lange weiter, bis das Programm übersetzt werden kann.

(b) Erstellen Sie die Datei `legalbutwrong.cc` mit folgendem Inhalt:

```
#include <iostream>

int main(int argc, char** argv)
{
  int n = 10;
  // calculate the sum of all numbers from 1 to n
  int i;
  int sum = 0;
  for (int j = 1 ; i <= n ; j = j+1)
  {
    sum = sum + j;
  }
  std::cout << sum << std::endl;
  return 0;
}
```

Kompilieren Sie das Programm und führen Sie es aus. Falls das Programm “hängt”, können Sie es mit der Tastenkombination “CTRL-C” beenden.

Das Programm ist zwar syntaktisch korrekt, aber es hat einen Fehler. Der Compiler kann Ihnen oft helfen, solche Probleme aufzuspüren. Hierzu müssen Sie ihn anweisen, Ihnen nicht nur Fehler, sondern auch Warnungen anzuzeigen. Sowohl GCC als auch clang benötigen hierfür die Option “-Wall” (warn all).

Kompilieren Sie das Programm erneut mit der zusätzlichen Option “-Wall” und beheben Sie die vom Compiler gemeldeten Probleme. Das Programm sollte nun die richtige Lösung (55) ausgeben.

Aufgabe 2: Mitternachtsformel

Schreiben Sie ein Programm, das von der Kommandozeile die Koeffizienten einer quadratischen Gleichung der Form $ax^2 + bx + c = 0$ abfragt und mit der Mitternachtsformel die beiden Nullstellen ausrechnet und ausgibt. Falls es keine eindeutige Lösung gibt ($a = b = 0$) oder die Lösung komplex ist, soll das Programm das abfangen und eine entsprechende Meldung ausgeben.

Hinweise:

- Um eine Kommazahl von der Kommandozeile einzulesen, verwenden Sie folgenden Codeschnipsel:

```
1  double v;
2  std::cout << "a = " << std::flush;
3  std::cin >> v;
```

- Um die Wurzel aus einer Zahl zu ziehen, gibt es die Funktion `sqrt`:

```
1  double root = std::sqrt(2.0);
```

  Bevor Sie diese Funktion verwenden können, müssen Sie am Anfang des Programms die Mathematik-Bibliothek von C++ mit der Zeile `#include <cmath>` einbinden.

Aufgabe 3: Collatz-Vermutung

Schreiben Sie eine Funktion `void collatz(int number)`, die folgendes tut:

- Gib `number` auf dem Bildschirm aus.
- Falls `number` gerade ist, teile die Zahl durch 2.
- Andernfalls multipliziere die Zahl mit 3 und addiere 1.
- Wiederhole diese Schritte, bis einer der folgenden Zahlenwerte erreicht wird:
  $1, 0, -1, -5$ oder -17.
- Gib abschließend dieses Ergebnis aus.

Hinweis:
Um herauszufinden, ob eine Zahl x gerade ist, testen Sie, ob $(x \mod 2) = 0$. Hierfür gibt es in C++ den Operator `x % y`, der den Rest der Ganzzahl-Division von x durch y berechnet:

```
1  int x = 23 / 5; // 4 (runden nach 0)
2  int y = 23 % 5; // 3 (vorzeichenbehafteter Divisionsrest)
```

Rufen Sie die obige Funktion aus der `main`-Funktion auf, wobei Sie den ersten Wert für `number` von der Tastatur einlesen. Auf diese Weise können Sie die entstehenden Zahlenfolgen für verschiedene Startwerte untersuchen. Warum kann der Wert 0 nur auf eine Weise erreicht werden? Und was haben alle Zahlen gemeinsam, die zum Wert 1 führen? Informieren Sie sich unter https://en.wikipedia.org/wiki/Collatz_conjecture über die mathematische Vermutung hinter diesen Zahlenfolgen.