

Oliver Gasser
Alexander Lochmann

Dortmund, den 20. November 2020

Praktikum zur Vorlesung

Einführung in die Programmierung

WS 20/21

Blatt 2

Es können 13 Punkte erreicht werden.

Grundlage: Ein- und Ausgabe auf der Konsole

Auf dem letzten Praktikumsblatt wurden bereits Zeichenketten und der Inhalt von Variablen auf der Konsole ausgegeben. Dies geschieht über den Ausgabestrom `std::cout`. Dabei können auch mehrere Zeichenketten und Variablen in einer Zeile ausgegeben werden. Ein Zeilenumbruch kann durch `std::endl` eingefügt werden. Dies sieht wie folgt aus:

```
1  int var1 = 1;
2  int var2 = 2;
3  std::cout << "Die Werte sind " << var1 << " und " << var2 << std::endl;
4  std::cout << "Dieser Text erscheint in der nächsten Zeile ";
5  std::cout << "und dieser Text erscheint in der gleichen Zeile" << std::endl;
```

Tastatureingaben können durch den Eingabestrom `std::cin` erfolgen. Im Gegensatz zu dem Ausgabeoperator `<<` muss der Eingabeoperator `>>` verwendet werden. Die Eingabe wird immer zeilenweise durchgeführt; d.h. sie wird mit der Taste `Enter` beendet.

```
1  int x;
2  std::cout << "Bitte geben Sie eine Zahl für x ein:";
3  std::cin >> x; // liest Zeichen ein, bis Enter gedrückt wird, und weist diese x zu
4  std::cout << "Die Zahl " << x << " wurde eingegeben." << std::endl;
```

Über die Direktive `using namespace std` wird der Namensbereich `std` verwendet. Da dieser auch den Ausgabestream `cout` sowie den Eingabestrom `cin` beinhaltet muss nach der `using`-Direktive die Kennzeichnung `std::` nicht mehr vor den Befehl gestellt werden. Das Gleiche gilt für den Zeilenumbruch `endl`. Dies sieht wie folgt aus:

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int y = 2;
6      cout << "y = " << y << endl;
7      return 0;
8  }
```

HINWEIS: Achten Sie darauf, dass Sie nur zulässige Eingaben machen. Wenn die einzulesende Variable z.B. vom Datentyp `int` ist, dann dürfen nur Ziffern eingegeben werden. Bei falschen Eingaben wird eine 0 ausgegeben und der Eingabestrom ist für alle folgenden Eingaben (zur Laufzeit) ungültig. Wie mit falschen Eingaben umgegangen werden kann, wird im späteren Verlauf der Vorlesung behandelt.

HINWEIS 2: Mit der Tastenkombination `Strg-C` können Sie eine laufende Instanz beenden.

Aufgabe 1: Fehlersuche (3 Punkte)

Ein Computer-Virus hat Ihren Rechner befallen und die Inhalte einiger Dateien verändert. Zufällig wurden Zeilen vertauscht und einzelne Zeichen gelöscht & verändert. Hier sehen Sie einen Ausdruck der Datei `widerstand.cpp`, die durch den Virus in Mitleidenschaft gezogen wurde.

```
using namespace std;
int main(){
\\ Dieses Programm berechnet den elektrischen Widerstand
double strom = 5.0
cout << "Berechneter Widerstand: ";
double widerstand = spannung / strom;
cout << widerstand >> endl;
double spannung = 230,0;
return 0; // fertig
}
#include <iostream>
```

Ihre Aufgabe besteht darin, die durch den Virus hervorgerufenen Änderungen rückgängig zu machen, so dass das Programm seinen ursprünglichen Zweck erfüllt. Legen Sie dazu einen neuen Ordner in Visual Studio Code mit dem Namen `Aufgabe_2_1` an: Gehen Sie dazu auf `Datei` und dann auf `Ordner zu Arbeitsbereich hinzufügen...` Erstellen Sie darin eine neue Datei mit dem Namen `widerstand.cpp`. Fügen Sie die ihnen zur Verfügung gestellte und vom Virus degenerierte Datei¹ `widerstand.cpp` hinzu. Stellen Sie anschließend deren Originalzustand wieder her, um ein lauffähiges Programm zu erhalten.

HINWEIS 3: Orientieren Sie sich dabei am Programmtext von Aufgabe 2 vom letzten Mal.

_____ (3)

Aufgabe 2: Geradengleichung (10 Punkte)

Aus der elementaren Algebra sind Geradengleichungen mit dem Muster $f(x) = mx + b$ bekannt. Es braucht lediglich zwei Punkte, um eine Gerade eindeutig zu beschreiben.

a) Legen Sie einen neuen Ordner mit dem Namen `Aufgabe_2_2a` und darin eine neue Datei mit dem Namen `gerade.cpp` an. Fügen Sie das Codegerüst aus der zur Verfügung gestellten Datei namens `gerade.cpp` hinzu. Ergänzen Sie den Code, indem Sie die Berechnung der Geradenparameter m, b unter Zuhilfenahme der Eingabevariablen `x1, y1, x2, y2`. Benutzen Sie dabei den Operator zur Division `/`. Die Werte für die Eingabevariablen `x1, y1, x2, y2` sollen über die Tastatur eingelesen und danach entsprechend zugewiesen werden. Nutzen Sie hierzu `std::cin`. Informieren Sie den Benutzer vor der Abfrage über eine geeignete Ausgabe, welche Zahl jeweils abgefragt wird. Details finden sich dazu am Anfang dieses Praktikumsblatts.

_____ (3)

b) Tragen Sie die Ergebnisse Ihrer Berechnung in folgender Tabelle ein. Lassen sich alle Geradengleichungen berechnen? Wenn nein, begründen Sie warum!

Punkte (x, y)	m	b
$(7, 8), (-3, -2)$		
$(4, 7), (1, -2)$		
$(2, 5), (-5, 5)$		
$(3, 2), (3, -1)$		

_____ (2)

¹<https://ls11-www.cs.tu-dortmund.de/teaching/ep2021sopra>

c) Erweitern Sie den Code so, dass zusätzlich überprüft werden soll, ob ein dritter Punkt auf der bereits berechneten Gerade liegt. Überprüfen Sie den Code, indem Sie zu allen berechneten Geraden aus Teil a) bestimmen, ob der Punkt (3, 4) auf der Gerade liegt. Geben Sie dazu die Meldung „Der Punkt (3, 4) liegt [nicht] auf der Geraden“ aus, je nachdem ob der Punkt auf der Geraden liegt oder nicht.

_____ (3)

d) Erweitern Sie den Code so, dass die Koordinaten des Punktes über den Eingabestrom mittels `std::cin` eingelesen werden. Informieren Sie den Benutzer vor der Abfrage über eine Ausgabe, welche Zahl jetzt abgefragt wird.

_____ (2)