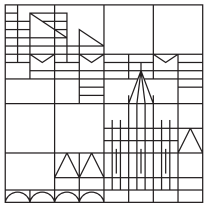
UNIVERSITÄT KONSTANZ
Fachbereich Physik (Theoretische Physik)
Dr. Stefan Gerlach
Raum P 817, Tel. (07531)88-3825
E-mail: stefan.gerlach@uni-konstanz.de

Universität Konstanz



Übungen zur Wissenschaftlichen Programmierung mit C++
Sommersemester 2024

Übungsblatt 1
Ausgabe 8.4., Übungen KW 15+16, Abgabe bis 21.4.

Einführung Wiss. Programmieren und C++

1. **Aufgabe: Wissenschaftliches Programmieren**

   (a) Wofür wird Software in der Wissenschaft genutzt? Nenne mindestens 5 Anwendungsbereiche und jeweils ein typisches Programm.

   (b) Welche Probleme kann es mit schlechter Forschungssoftware geben? Nenne 3 Beispiele.

   (c) Welche Herausforderungen gibt es bei Forschungssoftware? Nenne mindestens 3 Punkte.

   (d) Wie kann man die Nachhaltigkeit von eigener Forschungssoftware verbessern? Nenne 3 Punkte.

   (e) Nach welchen Kriterien sollte man die Programmiersprache wählen? Nenne 3 Punkte.

   (f) Warum eignet sich C++ gut für wissenschaftliche Software?

2. **Aufgabe: C++ Einführung**

   (a) Schreibe ein Hallo-Welt-Programm in C++. Kompiliere es und führe es aus.

   (b) Versuche durch Ändern des Hallo-Welt-Programms möglichst viele unterschiedliche Compilerfehler zun erzeugen.

   (c) Finde drei wichtige Konstanten, die in cmath definiert sind. Welchen Wert haben sie?

   (d) Definiere ein Makro für das Quadrat einer Zahl. Teste es anhand mehrerer Werte.

   (e) Warum sind die Klammern bei Makros wichtig? Probiere es aus.

   (f) Definiere und initialisiere eine Variable für jeden möglichen Datentyp und gebe die Werte aus.

   (g) Mit `sizeof` kann man die Größe eines Datentyps ausgeben. Welche Werte erhälst du für alle möglichen Datentyp?

   (h) Wie kommen die Minimal-/Maximalwerte der ganzahligen Datentypen zustande?

   (i) Was passiert bei einem **Überlauf** einer Variablen?

   (j) Definiere zwei Variablen und gebe deren Summe, Differenz, Produkt und Quotient aus.

   (k) Zeige den Unterschied von `i++` und `++i`.

   (l) Überprüfe mit dem Modulo-Operator, ob eine Zahl gerade ist.

(m) Definiere eine Variable und Teste auf `>0, <0`

(n) Wie kann man überprüfen, ob eine (ganze) Zahl gerade oder ungerade ist?

(o) Definiere zwei Variablen und Teste ob beide `>0`, beide `<0` oder eine von beiden `<0`

(p) Warum sollte man keine Fließkommazahlen vergleichen?

(q) Was passiert, wenn man ein `break` bei `switch` vergisst? Probiere es aus.

(r) Gebe die Zahlen von 1 bis 42 jeweils mit einer `for`, einer `while` und einer `do-while` Schleife aus.

(s) Der Laufindex einer Schleife kann auch eine Fließkommazahl sein. Probiere es für eine `for`-Schleife aus (Startwert=0, Endwert=1, Schrittweite=0.1). Welches Problem tritt dabei auf? Wie kann man es besser machen?

(t) Schreibe eine Schleife, die 3 äquidistante Werte zwischen 0 und 1 (inklusive) ausgibt. Verallgemeinere es dann auf N Werte.

(u) Schreibe eine Funktion, die "Hallo-Welt!" ausgibt

(v) Schreibe eine Funktion, die das Quadrat des Arguments berechnet und zurückgibt

(w) Schreibe eine Funktion, die den Quotienten zweier Zahlen berechnet und zurückgibt

(x) Schreibe eine rekursive und eine iterative Funktion zur Berechnung der Fakultät einer Zahl $n \in \mathbb{N}$. Teste die Funktionen für große n.

(y) Schreibe ein Programm, dass das Pascalsche Dreieck berechnet und ausgibt. Achtung: Die Formatierung ist nicht so einfach.

(z) Die Fibonacci-Folge ist gegeben durch

$$f_0 = 0, f_1 = 1, f_n = f_{n-1} + f_{n-2} (n > 1).$$

i. Schreibe ein Programm, dass die ersten Zahlen der Fibonacci-Folge ausgibt.

ii. Überprüfe, ob gilt

$$\sum_{i=0}^{n} f_i^2 = f_n f_{n+1}.$$

iii. Bestimme den Grenzwert $\lim_{n\to\infty} f_{n+1}/f_n$. Wie nennt man den Wert?

iv. Der gleiche Wert ergibt sich durch den Kettenbruch $1 + \frac{1}{1+\frac{1}{1+\ldots}}$. Stimmt das?

v. (*) Warum ist eine (simple) rekursive Implementierung der Fibonacci-Folge ungünstig? Wie sieht eine gute rekursive Implementierung aus?