1. Aufgabenblatt

Programmieren in C++ (Tutorium)

Grundlagen

1. Schreiben Sie ein Programm, das den Text `"Hello World!"` auf dem Bildschirm ausgibt.

2. Modifizieren Sie ihr Programm so, dass der Text `"Hello World!"` zunächst in einer Variable gespeichert wird. Deren Inhalt soll anschließend ausgegeben werden.

3. Modifizieren Sie ihr Programm erneut, so dass der Text `"Hello World!"` mithilfe der Präprozessor Direktive `#define` dem Symbol `HELLO` als Konstante zugewiesen und diese anschließend ausgegeben wird.

Einfache Datentypen und Operatoren

4. Schreiben Sie ein Programm, das das Volumen einer Party-Pizza (Kai's Pizza, Radius: 50 cm, Höhe: 1,2 cm) berechnet und auf dem Bildschirm ausgibt. Dazu soll der Wert `PI = 3.14159265` mittels `#define` als Konstante definiert und Radius, sowie Höhe der Pizza zunächst als Variablen `z` bzw. `a` gespeichert werden.

5. Stellen Sie die Zahlen $a = (155)_{10}$[1] und $b = (57)_{10}$ (per Hand, ohne Computer) im Binärsystem dar. Überlegen Sie sich zunächst auf dem Papier das Ergebnis der C++-Operationen `a<<3`, `b>>1`, `a^b`, `a&b`, `a|b` und `~b`. Schreiben Sie anschließend ein Programm, das Ihre Vorhersagen überprüft.

6. In C++, können Zahlen im Hexadezimalsystem durch die Schreibweise `0xHEX` dargestellt werden, wobei `HEX` die Darstellung der Zahl im Hexadezimalsystem ist.
   Schreiben Sie ein Programm, das die Anzahl der Sekunden/Minuten/Stunden in einem Jahr berechnet und jeweils auf dem Bildschirm ausgibt. Verwenden Sie in Ihrem Programm nur Zahlen in Hexadezimal-Darstellung (Sie müssen ggf. zunächst einige Umrechnungen per Hand vornehmen).

7. Abhängig vom Datentyp einer Variablen können Operatoren unterschiedliche Eigenschaften haben. Ein Beispiel ist der `"/"`-Operator, der bei ganzen Zahlen anders, als bei Gleitkomma-Zahlen wirkt.
   Schreiben Sie ein Programm, das den Wert von `a/2` bzw. `b/2` ausgibt, wobei a

[1] Die Darstellung $(\dots)_{10}$ wird verwendet, um eine Zahl als Dezimalzahl (Basis 10) zu kennzeichnen, wenn nicht offensichtlich ist, welches Zahlensystem gerade verwendet wird.

einen ganzzahligen Datentyp (z.B. `int`) und b einen Gleitkomma-Datentyp (z.B. `double`) hat, ansonsten aber $a = b$ ist. Worin besteht der Unterschied? Welche Wirkung hat der Operator `%` jeweils?

8. Verwenden Sie die Bibliothek `cmath`, die Sie mit `#include <cmath>`[2] einbinden können, um folgende Funktionen einer von Ihnen gewählten Gleitkommazahl x zu berechnen: $\sin(x) + \cos(x)$, e^x und $\log(|x|) + \sqrt{|x|}$.
   ZUSATZAUFGABE: Wie kann man x^7 berechnen, wenn man dazu nur die Logarithmus und Exponential-Funktion der cmath-Bibliothek zur Verfügung hat?

9. Mithilfe der Zeichencodierung ASCII (American Standard Code for Information Interchange) werden Zeichen als Folge von 7 BIT (z.B. A = 1000001) codiert[3]. Gleichzeitig können diese Bitfolgen natürlich auch als (Ganz-)Zahlen interpretiert werden.
   Schreiben Sie ein Programm, das die Zeichen Ihres Vornamens jeweils in eine Zahl umrechnet und deren Summe anschließend auf dem Bildschirm ausgibt. (Hinweis: Auf Seite 70 der Vorlesungsfolien haben Sie für die Umwandlung die Funktion `static_cast` kennengelernt.)

ergänzende Hinweise:
Auch wenn es -solange die Syntax stimmt- für den C++-Compiler egal ist, ob ein Programm schön geschrieben ist (was heißt das überhaupt?!), haben sich in der Praxis einige Richtlinien (s.g. Styleguides) für die Gestaltung eines Programms etabliert. Diese sollte man beachten, um es anderen Menschen zu erleichtern, den eigenen Programmcode zu lesen. Dazu gehört u.a.

- nur ein Programmstatement pro Zeile
  (Negativbeispiel: `int a = 5; a += 5; cout << a << endl;`)
- Der Programmcode sollte stets entsprechend der Schachtelungstiefe des Programms eingerückt werden. Eclipse erledigt das automatisch, wenn man zu Beginn einer Code-Zeile die TAB-Taste drückt.

```
// gut
int main()
{
   cout << 1 + 2 << endl;
   return 0;
}
```

```
// schlecht
int main()
{
cout << 1 + 2 << endl;
return 0;
}
```

- Leerzeichen um s.g. binäre (zweistellige) Operatoren (`a + b` anstatt `a+b`)
- keine Leerzeichen um s.g unäre (einstellige) Operatoren (`~x` statt `~ x`)

[2] Einige Beispiele zu Befehlen der `cmath`-Bibliothek finden Sie auf Seite 62 der Datei `Programmieren1-2.pdf`
[3] Auf dem Vorlesungsfolien finden Sie auf Seite 68 eine ASCII-Codierungstabelle.