Patrick Dammann, Philip Schill
Philip.Schill@iwr.uni-heidelberg.de
patrick@mathphys.fsk.uni-heidelberg.de

Übungsblatt 1

Deadline: 27.04.2015, 13:00 Uhr

Bemerkung:
Es müssen nur die Quelldateien der C++-Programme abgegeben werden, eine zusätzliche Text-Datei oder PDF-Datei ist **nicht** notwendig.

A00: Zeit messen (0 Punkte)

Notieren Sie sich die Zeit, die Sie zum Bearbeiten der einzelnen Aufgaben benötigen und vermerken Sie diese als Kommentar im Quellcode. Geben Sie zudem ein kurzes Feedback zu jeder Aufgabe (1-3 Sätze).
Dies fließt natürlich nicht in die Bewertung mit ein, sondern wird von uns dazu benutzt, die zukünftigen Übungsaufgaben zu verbessern.

A01: `int` und `float` (18 Punkte)

In dieser Aufgabe soll der Unterschied der Datentypen `int` und `float` verdeutlicht werden. Ergänzen Sie dazu das folgende C++-Programm.

```
1  int main()
2  {
3      int x1 = 7;
4      int y1 = 2;
5      int z1 = x1 / y1;
6
7      // Platziere deinen Code hier.
8
9      return 0;
10 }
```

(a) Legen Sie eine Quelldatei mit dem Namen `arithmetic.cpp` an und fügen Sie obigen Code ein. Geben Sie den Wert von `z1` mit Hilfe von `std::cout` aus.
Tipp: Vergessen Sie nicht, `#include <iostream>` am Anfang der Quelldatei einzufügen.

(b) Legen Sie drei Variablen `x2`, `y2`, `z2` vom Typ `float` an. Initialisieren Sie die Variablen mit `x2 = 7`, `y2 = 2` und `z2 = x2 / y2` und geben Sie den Wert von `z2` aus. Besitzen `z1` und `z2` denselben Wert? Überlegen Sie sich, wie dieses Ergebnis zustande kommt.

(c) Wiederholen Sie (b) und variieren Sie den Datentyp der drei Variablen zwischen `int` und `float`. Insgesamt gibt es 8 mögliche Kombinationen, bei einer fortlaufenden Nummerierung der Variablen enden Sie also mit `x8`, `y8`, `z8`.

Patrick Dammann, Philip Schill
Philip.Schill@iwr.uni-heidelberg.de
patrick@mathphys.fsk.uni-heidelberg.de

(d) Fügen Sie nach jedem `std::cout` einen Kommentar ein und beantworten Sie dort die folgenden Fragen:

- Was wurde ausgegeben?
- Wurde eine implizite Konversion von `int` nach `float` durchgeführt?
- Wurde eine implizite Konversion von `float` nach `int` durchgeführt?
- Welche Variable(n) war(en) von der Konversion betroffen?

(e) Legen Sie eine neue Datei `operators.cpp` an und kopieren Sie den Inhalt von `arithmetic.cpp` hinein. Verändern Sie das Programm, sodass der /=-Operator an Stelle des /-Operators benutzt wird. Mit dem /=-Operator können Sie das Ergebnis der Division direkt den Variablen `x1-x8` zuweisen, die Variablen `z1-z8` benötigen Sie nicht mehr.

Abgabe:

- `arithmetic.cpp` für Aufgabe (a)-(d)
- `operators.cpp` für Aufgabe (e)

A02: Finde die Fehler (18 Punkte)

In dieser Aufgabe soll geübt werden, wie man Fehler im Quellcode mit Hilfe des Compilers findet. Dazu finden Sie in Moodle ein Archiv mit Quelldateien. In jeder Quelldatei ist (genau) ein Fehler versteckt. Bearbeiten Sie die folgenden Aufgaben für jede Quelldatei.

(a) Versuchen Sie, den Quellcode zu kompilieren. Welche Fehlermeldung wird angezeigt? Kopieren Sie den wichtigen Teil der Fehlermeldung („Was ist der Fehler?“ und „In welcher Zeile ist der Fehler?“) als Kommentar in den Quellcode.

(b) Ergänzen Sie einen Kommentar, in welchem Sie in eigenen Worten erläutern, was die Meldung bzw. der Fehler bedeutet.

(c) Finden Sie den Fehler und beheben Sie ihn.

Halten Sie sich dabei an das folgende Format:

```
// urspruengliche Version der Zeile
// wichtiger Teil der Fehlermeldung (copy & paste)
// eigene Erklaerung der Meldung bzw. des Fehlers
korrigierte Programmzeile
```

Abgabe:

- Die Dateien 01.cpp bis 06.cpp mit den korrigierten C++-Programmen.

Patrick Dammann, Philip Schill
Philip.Schill@iwr.uni-heidelberg.de
patrick@mathphys.fsk.uni-heidelberg.de

A03: Gaußsche Osterformel (24 Punkte)

In dieser Aufgabe soll der Umgang mit arithmetischen Operatoren geübt werden. Dazu wird ein Programm erstellt, welches für ein gegebenes Jahr das Datum von Ostern berechnet. Die Berechnung erfolgt hierbei in mehreren Teilschritten. Bei dem folgenden Algorithmus ist x/y die Division und $[x]$ ist x abgerundet.

Input: Jahreszahl y.
$a \leftarrow y \mod 19$
$b \leftarrow y \mod 4$
$c \leftarrow y \mod 7$
$k \leftarrow [y/100]$
$p \leftarrow [(8k + 13)/25]$
$q \leftarrow [k/4]$
$M \leftarrow (15 + k - p - q) \mod 30$
$d \leftarrow (19a + M) \mod 30$
$N \leftarrow (4 + k - q) \mod 7$
$e \leftarrow (2b + 4c + 6d + N) \mod 7$
$x \leftarrow 22 + d + e$
if $x = 57$ **then**
    $x \leftarrow 50$
end if
if $x = 56$ **and** $d = 28$ **and** $a > 10$ **then**
    x = 49
end if
Output: Ostern ist am x. März.

Dabei steht der 32. März für den 1. April, der 33. März für den 2. April, usw.
Eine hervorragende Erklärung, was hinter der Osterformel steckt, findet sich bei `http://www.nabkal.de/gauss.html`.

Aufgabe:

(a) Legen Sie die Datei `ostern.cpp` an und implementieren Sie obigen Algorithmus. Lesen Sie mit Hilfe von `std::cin` eine Jahreszahl ein und berechnen Sie das Osterdatum für dieses Jahr. Geben Sie anschließend `Im Jahr Y ist Ostern am X. Maerz.` bzw. `Im Jahr Y ist Ostern am X. April.` aus.

(b) Testen Sie das Programm für 10 verschiedene Jahreszahlen und notieren Sie Ein- und Ausgabe als Kommentar im Quellcode.

Abgabe:

- `ostern.cpp` für Aufgabe (a) und (b)

Generelle Hinweise zur Abgabe

- Achten Sie auf eine korrekte Formatierung des Codes! Insbesondere müssen die Zeilen korrekt eingerückt sein. Bei Abgaben mit falscher Formatierung gibt es Punktabzug.

Patrick Dammann, Philip Schill
Philip.Schill@iwr.uni-heidelberg.de
patrick@mathphys.fsk.uni-heidelberg.de

- Jede Aufgabe muss auf Ihrem `ss15c###`-account im CIP-Pool kompilierbar sein. Lässt sich eine Aufgabe nicht kompilieren, so wird die Aufgabe als *nicht abgegeben* bewertet.
- Bearbeiten Sie jede Aufgabe in einem eigenen Ordner `A##` mit der Nummer der Aufgabe, zum Beispiel `A01`. Fassen Sie anschließend alle Ordner in einem Archiv zusammen und laden Sie dieses in Moodle hoch.
- Falls Sie einen eigenen Rechner benutzen, können Sie die Kompilierung testen, indem Sie mit dem Befehl `ssh ss15c###@pool.iwr.uni-heidelberg.de` eine ssh-Verbindung zu Ihrem Pool-Account herstellen. Mit dem Befehl `scp` können Sie Dateien von Ihrem Pool-Account auf Ihren Computer kopieren (oder umgekehrt).