

Informationstechnik

Übungsblatt 02

Institut für Technik der Informationsverarbeitung, Karlsruher Institut für Technologie

Zu Übung 01, Besprechung: Do., 02.06.2016, 14⁰⁰ - Neue Chemie

Aufgabe 2.01: Verständnisfragen

- a) Die Quelldatei wird zur Übersetzung an den _____ übergeben.
- b) Der _____ bindet eine Objektdatei mit anderen Modulen zu einer ausführbaren Datei.
- c) Beginnt die Zeile mit einem Doppelkreuz # am Anfang, so ist die Zeile für den _____ bestimmt.
- d) Ein Datentyp bestimmt
- die Art der Darstellung der Daten auf dem Bildschirm
 - die Art der internen Darstellung der Daten
 - die Anzahl der benötigten Speicherplätze in Bytes
- e) Wird eine interne Variable ohne Initialisierung definiert, so wird
- der Typ und Name der Variablen festgelegt
 - der Variablen automatisch ein Anfangswert zugewiesen
 - der entsprechende Speicherplatz für die Variable reserviert
- f) Zur Definition einer Variablen, die einmal initialisiert wird und später nicht mehr verändert werden soll, wird das Schlüsselwort _____ verwendet.
- g) In C++ ist der Index des **ersten** Arrayelements _____.
- h) In C++ ist der Index des **letzten** Arrayelements _____.
- i) Benennen Sie alle Fehler im folgenden Codeausschnitt:

```
float index = 0;  
float[5] myArray = {0, 1, 2, 3, 4, 5};  
myArray[index] = 10;
```

j) Nach Ausführung der folgenden Anweisungen speichert die Variable **x** den Wert:

```
int x = 1, y = 10;
if( y > 0 ) {
    if( y < 5 ) {
        x = 5;
    } else {
        x = 10;
    }
}
```

k) Nach Ausführung der folgenden Schleife speichert die Variable **x** den folgenden Wert:

- i. 1 ii. 2 iii. 3

```
int x = 0, y = 1;
while( ++y < 3 ) {
    x += y;
}
```

Aufgabe 2.02: Definition und Initialisierung von Variablen

Welche der folgenden Aussagen ist zulässig; das heißt sie wird vom C++ Compiler ohne Fehler übersetzt? Welche der Aussagen ist sinnvoll, also das Ergebnis erscheint so, wie man es vom Programmaufruf her erwartet.

Aussage	Zulässig	Sinnvoll	Bemerkung
<code>int a = 23;</code>			
<code>int b = 5.7;</code>			
<code>char c = 300;</code>			
<code>double f = 1.2E4;</code>			
<code>float d = 9 / 4;</code>			
<code>char g = '\\';</code>			
<code>double äquatum = 4;</code>			
<code>const float epi;</code>			
<code>float e = 9.0 / 4;</code>			
<code>short h = 32769;</code>			

Aufgabe 2.03: Gültigkeitsbereich

Gegeben sei der folgende Programmausschnitt:

```
double fq = 0.6180339887; //Fibonacci-Quotient
int b = 10000;

int hash( unsigned int x ) {
    double temp = x * fq - (int)( x * fq );
    return b * temp;
}
```

- In welchem Bereich des Programms kann auf die Variablen `fq`, `b`, `x` und `temp` zugegriffen werden?
- Welchen Wertebereich haben die Return-Werte der Funktion `hash()`?

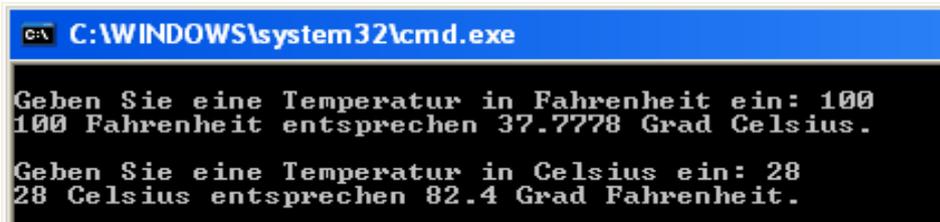
Aufgabe 2.04: Ein- und Ausgabe

Schreiben Sie ein C++-Programm, das einen Temperaturwert in der Einheit Fahrenheit im Dialog einliest und in die Einheit Celsius umrechnet und ausgibt.

Hinweis: Verwenden Sie zum Umrechnen die folgende Formel:

$$5 \cdot (\text{Fahrenheit} - 32) = 9 \cdot \text{Celsius}$$

Beispielausgabe:



```
C:\WINDOWS\system32\cmd.exe
Geben Sie eine Temperatur in Fahrenheit ein: 100
100 Fahrenheit entsprechen 37.7778 Grad Celsius.
Geben Sie eine Temperatur in Celsius ein: 28
28 Celsius entsprechen 82.4 Grad Fahrenheit.
```

Aufgabe 2.05: Mehrfachentscheidungen

Transformieren Sie den folgenden Codeausschnitt unter Anwendung einer **switch**-Anweisung und vermeiden Sie dabei den redundanten Code zur Ausgabe:

```
unsigned int number;
cout << "Geben Sie eine Zahl zwischen 1 und 5 ein: ";
cin >> number;

if( number == 0 ) {
    cout << "Eingabe zu klein!\n";
} else if( number == 1 ) {
    cout << "Eingabe ok und ungerade\n";
} else if( number == 2 ) {
    cout << "Eingabe ok und gerade\n";
} else if( number == 3 ) {
    cout << "Eingabe ok und ungerade\n";
} else if( number == 4 ) {
    cout << "Eingabe ok und gerade\n";
} else if( number == 5 ) {
    cout << "Eingabe ok und ungerade\n";
} else {
    cout << "Eingabe zu gross!\n";
}
```

Aufgabe 2.06: Anwendung von Kontrollstrukturen

Schreiben Sie die erforderlichen Anweisungen, um

- zwei Gleitpunktzahlen im Dialog einzulesen und die größere Zahl von der Kleineren zu subtrahieren und das Ergebnis auszugeben.
- Gleitpunktzahlen einzulesen und aufzusummieren, bis ihre Summe 100 überschreitet.
- positive Ganzzahlen solange einzulesen, bis eine negative Zahl eingegeben wird.
- den Anwender aufzufordern, drei verschiedene Ganzzahlen einzugeben. Die Aufforderung wird wiederholt, solange zwei von drei Zahlen übereinstimmen.

Hinweis: Bitte beachten Sie, dass Sie auch Schleifen nur mit Bedingungen und ohne Anweisungen schreiben können.

Aufgabe 2.07: Mehrdimensionale Arrays und Schleifen

Erstellen Sie ein C++ Programm, das im Dialog eine 3x3 Matrix einliest, ihre Determinante berechnet und das Ergebnis auf dem Bildschirm ausgibt. Verwenden Sie zur internen Speicherung der Matrix ein mehrdimensionales Array.

Hinweis:

$$\det A = \det \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$
$$= a_{11}a_{22}a_{33} + a_{12}a_{23}a_{31} + a_{13}a_{21}a_{32} - a_{13}a_{22}a_{31} - a_{12}a_{21}a_{33} - a_{11}a_{23}a_{32}.$$

Beispielausgabe:

```
C:\Windows\system32\cmd.exe
Zeile 1, Spalte 1: 11
Zeile 1, Spalte 2: 12
Zeile 1, Spalte 3: -13
Zeile 2, Spalte 1: 21
Zeile 2, Spalte 2: -22
Zeile 2, Spalte 3: 23
Zeile 3, Spalte 1: -31
Zeile 3, Spalte 2: 32
Zeile 3, Spalte 3: 33

Eingegebene Matrix:
11      12      -13
21     -22       23
-31     32       33

Die Determinante betraegt: -32824
```