

Programmiertechnik IKA 9/13 – Übungsblatt für die ersten Versuche (2)

Lösen Sie die folgenden Aufgaben OHNE Hilfe der „älteren“ IKA-Kollegen...

Aufgabe 1a:

Schreiben Sie ein Programm, das vom Benutzer den Vornamen (ohne Leerzeichen) erfragt und anschließend mit Text davor und dahinter ausgibt, etwa so:

```
Guten Tag, wie heisst Du? Klaus  
Hallo Klaus, wie gehts?
```

[Hinweis auch für alle folgenden und zukünftigen Beispiel-„Screenshots“: Ich versuche, die vom Benutzer eingegebenen Zahlen, Zeichen, Texte entweder farblich und/oder durch fette kursive Schrift von den in normaler Schrift dargestellten Ausgaben des Programms zu unterscheiden. Je nach Drucker / Kopierer kann das Ergebnis unterschiedlich ausfallen...]

Aufgabe 1b:

Das Doppel-s und das fehlende Apostroph stören uns! Es sollte da ja eigentlich stehen:

```
Guten Tag, wie heißt Du? Klaus  
Hallo Klaus, wie geht`s?
```

Mhm – wie könnten wir das hinbekommen???

Kleiner Tipp: der ASCII-Wert für das „Ligatur-sz“ (ß) ist (in C++ für die Windows-Konsole) dezimal 225, das Apostroph („Auslassungszeichen“) hat den ASCII-Code dezimal 39.

Aufgabe 1c:

Geben Sie zum Spaß bei der Frage nach Ihrem Namen Vor- und Nachname – getrennt durch das Leerzeichen – ein! Was stellen Sie fest?

Welcher Befehl in Ihrem Programm ist für dieses Verhalten verantwortlich? Wie wird er also wohl intern funktionieren? Notieren Sie hier Ihre Überlegungen dazu:

Probieren Sie auch einmal die beiden unterschiedlichen Varianten gegen Ende des Programms aus und ziehen Sie den beobachtbaren Effekt in Ihre Überlegungen ein:

```
// Variante 1:  
system(„pause“);  
// Variante 2:  
getchar();  
getchar();
```

Aufgabe 2a:

Deklariieren Sie für alle Standard-Datentypen (elementare Basis-Datentypen) je eine Variable OHNE Wertzuweisung, zum Beispiel **MeinInt** für ein Ganzzahldatum.

Lassen Sie nun mit dem sizeof-Operator den für die Daten verbrauchten Speicherplatz am Konsolenbildschirm ausgeben.

Muster für den Gebrauch des sizeof-Operators: ... **sizeof(MeinInt)** ...

Aufgabe 2b:

Erweitern Sie den Quellcode um erweiterte Basisdatentypen, zum Beispiel **long int**, **short int** oder **long long int**. Probieren Sie, welche Erweiterungen für welche Basisdatentypen funktionieren oder nicht.

Aufgabe 2c:

Weisen Sie der „short int“-Variablen den Wert 32000 zu und lassen sich den Wert auf der Konsole anzeigen.

Addieren Sie den Wert 1000 zu diesem Datum und lassen es erneut anzeigen.

Erklären Sie, was jetzt passiert ist...

Aufgabe 2d:

Es gibt noch die Erweiterung **unsigned**, also eine Zahl ohne Vorzeichen (nur positive Werte).

Probieren Sie, für welche Basisdatentypen die „unsigned“-Erweiterung keine Fehler erzeugt.

Setzen Sie nun den Wert einer „positiven kurzen Ganzzahl“ auf 65000 und wiederholen das Experiment aus Aufgabe 2c: also den Wert 1000 addieren und den neuen Wert anzeigen.

Stimmt Ihre Erklärung für den Effekt in Aufgabe 2c noch?

JETZT ABER ERST MAL PAUSE....

Aufgabe 3:

Lösen die noch offenen Aufgaben aus dem letzten Praktikum...

Aufgabe 4:

Eine „Freestyle“-Aufgabe: Könnte man mit dem bisherigen Wissen schon andere praktische Fragestellungen lösen?

Notieren Sie Ihre Ideen und versuchen Sie selbst, diese Fragestellungen zu lösen.

Ach, ich denke zum Beispiel an folgende Fragestellung:

Wie viele Stimmen haben die Piraten bei DIESER Wahl bekommen:	23000
Wie viele Stimmen haben die Piraten bei der LETZTEN Wahl bekommen:	20000
Das ist eine Veraenderung um 15 Prozent	

Notieren Sie auch, wo es Probleme gibt und was nicht so recht funktioniert...

Wir haben ja noch genügend Zeit, tiefer in die Materie einzusteigen ☺