

Zu Übung 02, Besprechung: Do., 14.06.2018, 14⁰⁰ – Neue Chemie**Aufgabe 3.01: Verständnisfragen**

- a) Ein Funktionsaufruf ist ein Ausdruck, dessen Typ bestimmt ist durch
- die an die Funktion übergebenen Argumente.
 - die im Funktionskopf deklarierten Parameter.
 - den Return-Wert der Funktion.
- b) Der Prototyp einer Funktion stellt dem Compiler Informationen über
- den Return-Typ der Funktion bereit.
 - die Namen der Parameter bereit.
 - den Typ jedes Parameters bereit.
- c) Ein Compiler erkennt eine falsche Anzahl von Argumenten nicht.
Richtig / Falsch
- d) Ein außerhalb einer Funktion definierte Variable wird als _____ bezeichnet.
- e) Eine Funktion kann innerhalb einer anderen Funktion definiert werden.
Richtig / Falsch
- f) Bei der Ausführung der Anweisungen:
- ```
string name;
cin >> name;
```
- werden Zeichen von der Standardeingabe eingelesen und zwar
- alle Zeichen einer Zeile ohne führende Zwischenraumzeichen
  - genau ein Wort ohne führende Zwischenraumzeichen
  - eine ganz Textzeile
- g) Zum Verketteten zweier Objekte vom Typ `string` kann man den Operator \_\_\_\_\_ verwenden.
- h) Das erste Zeichen in einem String hat die Position \_\_\_\_\_.
- i) Für den Zugriff auf die einzelnen Zeichen in einem `string`-Objekt kann der Operator \_\_\_\_\_ verwendet werden.

### Aufgabe 3.02: Funktionen – Deklaration, Prototypen, Aufruf

a) Bestimmen Sie die Fehler in folgenden Prototypen:

- i. `double calculate double x, double y;`
- ii. `void myFunc( int n, m );`
- iii. `int your-Func();`
- iv. `Bool test( void );`

b) Welche der folgenden Funktionsaufrufe sind korrekt? Wenn der Funktionsaufruf nicht korrekt ist, beschreiben Sie den Fehler.

- i. `int max( int a, int b, int c );`  
`int result = max( 7, 12 );`
- ii. `double square( double wert );`  
`double x = 2.1;`  
`cout << square( x );`
- iii. `int random( void );`  
`random( 1 );`
- iv. `int random( int a );`  
`random( 1 );`

### Aufgabe 3.03: Referenzen

a) Was gibt das folgende Programm auf der Konsole aus? Bitte überlegen Sie, bevor Sie das Programm direkt kompilieren und sich die Ergebnisse anzeigen lassen.

```
#include <iostream>
using namespace std;

int main() {
 int a = 3;
 int& b = a; //Referenz zu a
 cout << b << "\n";
 a = 18;
 cout << b << "\n";
 b = 25;
 cout << a << "\n";
}
```

b) Welche Fehler werden im folgenden Programm gemacht?

```
#include <iostream>
using namespace std;

int main() {
 int& c = 3;
 bool d = false;
 int& e = d;
}
```

### Aufgabe 3.04: Referenzen und Zeiger

- c) Beschreiben Sie jeweils die Aufgabe der entsprechenden Zeilen des unten stehenden Programms.
- d) Was erzeugt das Programm für eine Ausgabe. Bitte überlegen Sie, bevor Sie das Programm direkt kompilieren und sich die Ergebnisse anzeigen lassen.
- e) Welche zwei Besonderheiten fallen Ihnen in Bezug auf die beiden Zeilen `++ptr` und Folgende auf.

```
#include <iostream>
using namespace std;

int main() {
 int var = 256;
 int& ref = var;
 int* ptr = &var;

 //Referenzen
 ++ref;
 cout << var << "\t\t" << ref << endl;

 var += 64;
 cout << var << "\t\t" << ref << endl;

 cout << &var << "\t" << &ref << endl << endl;

 //Zeiger
 ++var;
 cout << var << "\t\t" << ptr << "\t" << *ptr << endl;

 *ptr += 12;
 cout << var << "\t\t" << ptr << "\t" << *ptr << endl;

 cout << &var << "\t" << &ptr << endl << endl;

 ++ptr;
 cout << var << "\t\t" << ptr << "\t" << *ptr << endl;

 return 0;
}
```

### Aufgabe 3.05: Arrays und Zeigerarithmetik

Was erzeugt das folgende Programm für eine Ausgabe auf dem Bildschirm:

Hinweise: - `sizeof()` gibt die Anzahl an Bytes des übergebenen Arrays zurück.

```
#include <iostream>

using namespace std;

int main() {
 char arr[] = "Informationstechnikbuch"; //C-String: Letztes Zeichen '\0'
 char* p = arr;
 char* q = arr + 6;

 cout << q - p - 3 << *q;
```

```
p = q++ + 8; //Post-Inkrement von q: Erst Zuweisung p = q + 8, dann
 //q=q+1;

while(p < q + 9) {
 cout << *p++;
}

cout << *q << *(p - 3) << *(arr + 4);

p = arr + 19;
cout << *p << *--q << *(arr + sizeof(arr) - 2) << "nen";

return 0;
}
```

### Aufgabe 3.06: Strings

Schreiben Sie die Funktion `palindrom`, die feststellt, ob der gegebene String `str_pali` ein Palindrom ist. Ein Palindrom kann man von vorne und von hinten lesen. Jedesmal ist der String gleich. Beispiel: Rentner.

Der String `str_pali` wird in `main` definiert.

Vorbereitung: Wie sieht die Eingabe- und Ausgabeschnittstelle zur Funktion `palindrom` aus?