



Note / normierte Punkte

# Klausur in Programmieren

Sommer 2013, 15. Juli 2012  
Dauer: 1,5h  
Hilfsmittel: Keine (Wörterbücher sind auf Nachfrage erlaubt)

**Name:**  
**Matrikelnr.:**

Aufgabe	1	2	3	4	5	6	Summe
Punkte max	8	15	20	19	23	15	100
Punkte							

*Alle Fragen beziehen sich auf den Stoff der Vorlesung. Somit sind sie z.B. bezogen auf die Programmiersprache C++. Auch sonst gelten die Konventionen wie in unserer Vorlesung.*

## 1. Aufgabe: Grundlagen

a) Geben Sie an, welcher der C/C++ Ausdrücke eine Variablendefinition (F) und welcher eine Variablendeklaration (D) ist. (4 P):

- 1) double d = sin(0.12); .....         F
- 2) char c = 'B'; .....         F
- 3) int d; .....         D
- 4) cout << "Definition: "; .....

b) Welche Werte enthalten die nachfolgenden Variablen: (4 P)

- 1) int i = 13.3 \* 3.0; i ==         39
- 2) double d = 1.5 \* 5; d ==         7.5
- 3) int i = 5;  
double d = ( (i++ \* ((i + 3) % 3)) - 1) / 2.0; i ==         6        , d ==         4.5

## 2. Aufgabe: Grundlagen

a) Schreiben Sie ein vollständiges und lauffähiges Hauptprogramm, das 10 ganze Zahlen von der Konsole einliest und daraus den Mittelwert berechnet. Dieser Wert soll dann in einer Variablen mit geeignetem Datentyp zwischenspeichert und danach auf die Konsole ausgegeben werden. (9 P)

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

int main()
{
    int iSumme = 0;
    for(int li=0; li < 10; ++li)
    {
        int iWert;
        cout << "Wert " << li+1 << ": ";
        cin >> iWert;
        iSumme += iWert;
    }
    double dMittelwert = (double)iSumme / 10.0;
    cout << "Mittelwert: " << dMittelwert << endl;

    return 0;
}
```

b) Welche Schleifen gibt es in C/C++? Geben Sie kleine Beispiele: (6 P)

while:

```
double dValue = 1.0;
while(dValue < 100.0)
{
    dValue *= 10.0;
    if(dValue > 40.0)
    {
        dValue /= 2.0;
    }
}
```

do .. while:

```
int iValue = 20;
do
{
    iValue *= 4;
}
while(iValue < 50;
```

for:

```
for(int li=0; li<10; ++li)
{
    cout << li << endl;
}
```

### 3. Aufgabe: Funktionen

a) Erklären Sie aus welchen Teilen eine Funktion besteht (strukturiert vorgehen!) (14 P):

#### 1. Funktionskopf

- a.) Datentyp des Rückgabewertes
  - void bedeutet kein Rückgabewert
- b.) Funktionsname
- c.) Parameterliste in runden Klammern eingeschlossen
  - Parameterliste besteht aus keinem, einem oder beliebig vielen Parametern
  - mehrere Parameter werden mit einem Komma getrennt
  - ein Parameter wird wie eine Variable deklariert und verwendet

#### 2. Funktionsrumpf

- entspricht einem Anweisungsblock
- Anweisungsblöcke enthalten keine oder beliebig viele mit einem Semikolon getrennte Anweisungen und beginnen mit '{' und enden mit '}'.
- mit 'return' wird ein Funktionsergebnis zurück gegeben

b) Wie bezeichnet man bei folgenden Beispielen die Art, mit der die Parameter der Funktion übergeben werden? Erklären Sie den Mechanismus in kurzen Stichworten (6 P):

i. `int function1(double& dValue)`

Aufruf mittels call by reference

Es wird die Variable selbst als Referenz übergeben. Änderungen dieses Parameters innerhalb der Funktion bedeuten auch Änderungen der übergebenen Variablen.

ii. `void function2(int iValue)`

Aufruf mittels call by value.

Es wird eine Kopie des übergebenen Wertes im Parameter gespeichert. Änderungen des Parameters haben keinen Einfluss auf Variablen im Funktionsaufruf.

- Lösen Sie die Aufgaben bitte auf dem Blatt -

## 4. Aufgabe: Array/Feld, Indizierung

a) Erklären Sie den Unterschied zwischen einem Array und einer Struktur. (4 P)

Ein Array ist ein zusammengesetzter Datentyp, bei dem alle Elemente denselben Datentyp besitzen.

Eine Struktur ist ein zusammengesetzter Datentyp, bei dem die Elemente unterschiedliche Datentypen besitzen können.

b) Definieren Sie eine Variable `adFeld`, der ein dynamisch angelegtes Array der Größe 20 mit einem Felddatentyp `double` zugewiesen wird. Weisen Sie dem Feldelement von `adFeld` mit Index 5 den Wert 7.5 zu und weisen Sie anschließend dem Feldelement mit Index 6 den Inhalt des Feldelements mit Index 5 zu (damit ist nicht der konstante Wert 7.5 gemeint!). Schreiben Sie eine Schleife, die nur die Werte des Arrays in jeweils einer eigenen Zeile auf die Konsole ausgibt (sie können voraussetzen, dass die Bibliothek `iostream` bereits eingebunden wurde). Geben Sie als letzten Schritt das dynamisch angelegte Array wieder frei. (10 P)

```
double* adFeld = new double[20];

adFeld[5] = 7.5;
adFeld[6] = adFeld[5];

for(int li=0; li < 20; ++li)
{
    std::cout << adFeld[li] << std::endl;
}

delete [] adFeld;
```

c) Deklarieren Sie eine Struktur mit dem Namen `sValues`. Die Struktur soll eine `double` und eine `int` Variable besitzen. Legen Sie eine Variable mit dem Datentyp `sValues` an und weisen Sie der `double`-Komponente den Wert 5.4 und der `int`-Komponente 3 zu (kein weiterer Code als der geforderte! Ein- Ausgaben oder `main`-Funktion geben Punktabzug!) (5 P):

```
struct sValues
{
    double dVar1;
    int     iVar2;
};

sValues values;
values.dVar1 = 5.4;
values.iVar2 = 3;
```

- Lösen Sie die Aufgaben bitte auf dem Blatt -

## 5. Aufgabe: Zeichenketten

a) Schreiben Sie eine Funktion `strlen`, die die Länge einer mit 0 terminierten Zeichenkette bestimmt. Die Zeichenkette soll als Parameter übergeben werden. Der Funktionswert soll die Länge zurückgeben (kein Hauptprogramm, keine Ein- oder Ausgabe!). (8 P)

```
int strlen(char* acInString)
{
    int iIndex = 0;
    while(acInString[iIndex] != 0)
    {
        iIndex++;
    }
    return iIndex;
}
```

b) Schreiben Sie eine Funktion `replace`, der eine Zeichenkette `acString` und zwei Zeichen `cSource` und `cTarget` als Parameter übergeben wird. Alle Zeichen von `acString` die gleich dem Zeichen `cSource` sind, sollen durch das Zeichen `cTarget` ersetzt werden. (kein Hauptprogramm, keine Ein- oder Ausgabe!). (15 P)

`replace("Ersetzen", 'e', 'u') → "Ersutzun"`

```
void replace(char* acString, char cSource, char cTarget)
{
    int iIndex = 0;
    while(acString[iIndex] != 0)
    {
        if(acString[iIndex] == cSource)
        {
            acString[iIndex] = cTarget;
        }
        iIndex++;
    }
}
```

## 6. Aufgabe: Algorithmus

Was macht die nachfolgende Funktion unknown? Was wird auf die Konsole ausgegeben?

Bitte beschreiben Sie die Funktionsweise möglichst abstrakt – Romane geben Abzug! (15 P)

```
#include <iostream>

using namespace std;

void unknown(char* acUnknown)
{
    int iVar1 = 0;
    while(acUnknown[iVar1] != 0)
    {
        iVar1++;
    }

    int iVar2 = 0;
    while(iVar2 < iVar1/2)
    {
        char cTmp = acUnknown[iVar2];
        acUnknown[iVar2] = acUnknown[iVar1 - iVar2-1];
        acUnknown[iVar1 - iVar2-1] = cTmp;
        iVar2++;
    }
}

void main()
{
    char acVar[] = "WhatsThat?";
    unknown(acVar);

    cout << "Ergebnis: " << acVar << endl;
}
```

Auf die Konsole wird ausgegeben: "?tahTstahW"  
Die Funktion unknown dreht die übergebene Zeichenkette um.