

Täuschungsversuch

Liebe Studierende des Fachbereiches Informatik,

in letzter Zeit häufen sich wieder einmal die Täuschungsversuche. Die Linie des Fachbereiches hierzu ist auch im Interesse der ehrlichen Studierenden sehr klar.

Mit einem Täuschungsversuch fallen Sie definitiv (ggf. sogar nachträglich) durch. Unter Umständen beenden Sie mit dem Täuschungsversuch Ihr Studium.

Durch gegenseitiges Abschreiben ('Ich habe ihm nur meine Unterlagen gegeben, um zu helfen') fallen übrigens beide Beteiligten durch.

Themen in der Klausur

Zeiger- und Wertesemantik (Operator * und &)

Speicherbelegung auf Heap- und Stackspeicher veranschaulichen

Call by Value, Call by Reference (Werte- und Referenzübergabe von Parameter an Funktionen)

Konstruktor, Destruktor, Was gibt das Programm aus?

**Schreiben Sie einen Tests / Tests ... für eine
Funktion PlusMinus ...**

Themen in der Klausur (2)

Kopier-Konstruktor, Verschiebe-Konstruktor (&&)

Zuweisungs-Operator, Verschiebe-Zuweisungsoperator (&&)

Konstante Methoden, konstante Variablen,
STL-Container (list, vector), Iteratoren, einfache STL-Algorithmen
(max_element, for_each, find, sort ...)

Intelligente Zeiger: unique_ptr, smart_ptr, weak_ptr, kann man super mit
Speicherbelegung zeichnen und „was gibt folgendes Programm
aus?“ kombinieren.

Lambda-Ausdrücke

Es muss vom Kopf in die Finger, die Zeit wird das Problem in der Klausur sein, die meisten Aufgaben werden sie schaffen, aber schaffen Sie das auch in 90 Minuten unter Klausurbedingungen, wenn das Blut im Kopf pocht?

Deshalb; üben am Rechner und Clicker-Aufgaben auf Folien
Aufgaben haben Sie genug aus alten Klausuren und aus den Übungen!

Clicker: Anzahl Schleifendurchläufe; Wert von zaehler

```
void loop1() {  
    int zaehler = 0;  
    for (int i=12; i < 24; ++i) {  
        zaehler ++;  
    }  
}
```

1. 6
2. 11
3. 12
4. 13

Ergebnis:

___ 1 ___ 2 ___ 3 ___ 4

Clicker: Anzahl Schleifendurchläufe; Wert von zaehler

```
void loop1() {  
...  
    int zaehler = 0;  
    for (int i=N+2; i < K; i++) {  
        zaehler++;  
    }  
}
```

1. $K - (N+2)$
2. 11
3. $N+2 - K$
4. $N+K+2$

Ergebnis:

 1 2 3 4

Klausurvorbereitung

Allgemein

BildschirmAusgabe: letztes Print

```
void Print(int j) {  
    cout << j << " ";  
}  
int main() {  
    list<int> cont;  
    cont.push_back(3);  cont.push_back(9);  
    cont.push_back(3);  cont.push_back(8);  
    cont.push_back(7);  
    // Ausgabe ist 3 9 3 8 7  
    for_each(cont.begin(), cont.end(), Print);  
  
    remove(cont.begin(), cont.end(), 3);  
    for_each(cont.begin(), cont.end(), Print);  
}
```

Ausgabe ?

1. 4 9 3 8 7
2. 9 8 7
3. 9 8 7 8 7
4. 9 8 7 3 3

Ergebnis

```
___ 4 9 3 8 7      ___ 9 8 7  
-_- 9 8 7 8 7      ___ 9 8 7 3 3
```

Übung 7: Figur, Kreis, Ring

```
class Figur {
    int dummy;
public:
    Figur() {dummy=0; datei << "+F";}
    ~Figur() {datei << "-F";}
    void Umfang() const
        {datei<<"F"<< 0;}
    virtual void Flaeche()const=0;
};
class Ring: public Kreis {
    int rRad; // Innenkreis-Radius
    /* ...*/
};
```

1. Bedeutung von **const**

2. Include-Wächter zu Figur

```
class Kreis: public Figur {
protected:
    enum {PI=3};
    int rad; // Außenkreis-Radius
public:
    Kreis(): Figur(8), rad(4) {
        datei << "+K" << rad << " ";
    }
    Kreis(int r): Figur() {
        rad=r;
        datei << "+K" << rad << " ";
    }
    ~Kreis() {
        datei << "-K" << rad << " ";
    }
    /* ...*/
};
```

Übung 7: Include-Wächter von Figur

```
#ifndef FIGUR_HEADER
#define FIGUR_HEADER
class Figur {
    int dummy;
public:
    Figur() {dummy=0; datei << "+F";}
    ~Figur() {datei << "-F";}
    void Umfang() const
        {datei<<"F"<< 0;}
    virtual void Flaeche()const=0;
};
class Ring: public Kreis {
    int rRad; // Innenkreis-Radius
    /* ...*/
};
#endif
```

Klausurvorbereitung

Auswahl an Themen

Auswahl an Themen, für die ich Aufgaben habe

1. Zeiger allgemein (Was bedeutet: &variable, *zeiger etc.?)
2. Speicherbelegung auf dem Heap und auf dem Stack veranschaulichen (mit +Hund/-Hund)
3. Objekterzeugung, Konstruktor/Destruktor (+Hund –Hund)
4. Intelligente Zeiger
5. STL
6. Polymorphie (virtual) Dynamischer Typ / Statischer Typ
7. Testen
8. Was habe ich vergessen?

Auf Homepage stehen alle meine alten Klausuren



[Alte Klausuren von mir mit Lösungen seit 2001](#)

Nützliche Hilfen

[Lehrbücher zu C++ \(12.08.2023\)](#)

[Einführung: Benutzung von Visual Studios, \(13.07.2024\)](#)

[Hilfen zu speziellen Problemen mit Visual Studio und C++, \(13.07.2024\)](#)

[Visual Studios 2022 Comm Edition installieren und nutzen, \(13.07.2024\)](#)

[Links zu Subversion / Tortoise](#)

Fachbereich Informatik
■ C++-Programmierparadigmen
Klausuren zu Programmierkonzepten
Informatik III



Fachhochschule
Braunschweig/Wolfenbüttel

Hon. Prof. Dr.-Ing. Hartmut Helmke
Deutsches Zentrum für Luft-
und Raumfahrt e.V. (DLR)
Institut für Flugführung
Abteilung Lotsenassistenzsysteme
Postfach 32 67
38108 Braunschweig

Email: Hartmut.Helmke@DLR.DE
Tel. 0531 295-2599

**Klausuren zu Vorlesungen von Hon. Prof. Dr. Helmke mit
Schwerpunkt C++**

SS 2001

fali.

Ich werde auch Fragen
Dazu per E-Mail beantworten
Kann ein paar Tagen dauern.

WS 2023/24

[Klausur Aufgaben](#) (19.1.2024) [Klausur Lösungen](#) (19.1.2024)

WS 2024/25

[Klausur Aufgaben](#) (17.1.2025) [Klausur Lösungen](#) (17.1.2025)

Letzte Änderung 17.01.2025