

## Clicker

Bitte den Link

<https://vc2.sonia.de/b/har-2cy-qhv-bj0>  
für WS 2025/26 nutzen.

# Aufgabe 01

joh.katron@ostfalia.de

# Termine des Semesters

Termin		Vorlesung	Übungen und Feedback
Woche		Freitag ; Block 1+2	
5	24. Okt	Werte-, Zeigerparameter etc., Klassen string, vector<T> ADT; Klassen, tiefe und flache Kopie	
6	31. Okt	<b>fällt aus: Feiertag</b>	Finale Abgabe <b>Fr. 31.10</b> Übung 2; Demo per BBB vorher
7	07. Nov	tiefe und flache Kopie: Kopierkonstruktor, LogTrace ; Operatoren	7.11; Teambildung abgeschlossen
8	14. Nov	Operatoren (Teil 2); Templates, Polymorphie	
9	21. Nov	STL, Iteratoren;	
10	28. Nov	lineare und assoziative Container; STL Algorithmen	<b>Bis 27.11 Erste</b> Teamaufgabe abgegeben; Demo per BBB vorher
11	05. Dez	Algorithmus oder Methode Klasse unique_ptr,, shared_ptr, Lambda-Ausdrücke	
12	12. Dez	Verschiebeoperatoren ,	
13	19. Dez	Rest; Vorbereitung Klausur	Finale Abgabe Teamaufgabe <b>Mi. 17.12</b> ; Demo per BBB vorher

Klausur: **noch offen** Uhr  
 Klausureinsicht: Jan 2026 **xxx** Uhr

## Rückblick

Foliensatz C1\_Quellcodeorganisation.pptx mit Namespace

Demo: Aufteilung von Code auf Source, Header-Dateien

Motivation von Zeigern und new und delete  
(dynamischerVektorMatrix.pptx)

Motivation von Zeigern, (int\*, &var, \*var, int&..)  
new und delete

Demo: struct Vector erweitert, sodass nun dynamische Größen möglich  
sind (Methode Resize ganz zum Schluss)

## Clicker-“Abstimmung“

```
void funkWert(int i) { i = 127;}  
void funkPtr(int* i) { *i = 64;}  
void funkRef(int& i) { i = 888;}  
  
void main(){  
    int j=11; funkWert(j); cout << j;  
    j = 11; funkPtr(&j); cout << j;  
    j = 11; funkRef(j); cout << j;  
}
```

Was wird auf dem Bildschirm ausgegeben  
(Ignorieren von Blanks) ?

- |    |     |    |     |
|----|-----|----|-----|
| 1. | 11  | 11 | 11  |
| 2. | 11  | 11 | 888 |
| 3. | 127 | 64 | 888 |
| 4. | 11  | 64 | 888 |
| 5. | 127 | 11 | 888 |

## Clicker-“Abstimmung“

```
void funkWert(double d) { d = 127;}  
void funkRef(double& d) { d = 888;}  
  
void main(){  
    double j=11; funkWert(j); cout << j;  
    j = 11; funkRef(j); cout << j;  
}
```

Was wird auf dem Bildschirm ausgegeben?

1. 11                      11
2. 11                      888
3. 127                     888
4. 127                     11

## Clicker-“Abstimmung“

```
struct Student {  
    int alter;  
};  
  
void fWert(Student s) {  
    s.alter = 127;  
}  
  
void fPtr(Student* s) {  
    (*s).alter = 888; // bzw. s->alter = 888;  
}  
  
void main(){  
    Student s1; s1.alter = 11;; fWert(s1); cout << s1.alter;  
    s1.alter = 11; fPtr(&s1); cout << s1.alter;  
}
```

Was wird auf dem Bildschirm ausgegeben?

1. 11            11
2. 11            888
3. 127          888
4. 127          11

## Clicker-“Abstimmung“

```
int main(){
  int N=5;
  int N1=16
  int arr[100] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};
  arr[-1] = 12; arr[-4] = 12;
  for (int i=1; i < N; ++i) {
    cout << arr[i] << " ";
  }
}
```

Welche Bildschirmausgabe könnte das nebenstehende Programm erzeugen?

Was wird auf dem Bildschirm ausgegeben?

1. 2 3 4 5
2. 1 2 3 4 5
3. 2 3 4 5 6
4. 2 3 4 5 6 7 8 9 10 0 0

<https://cppinsights.io/>

## Transformation von abgekürzten Code in vollständigen Code

insight:

```
1 #include <cstdio>
2
3 int main()
4 {
5     const char arr[10] = {2, 4, 6, 8, '\0', '\0', '\0', '\0', '\0', '\0'};
6     {
7         char const (&__range1)[10] = arr;
8         const char * __begin1 = __range1;
9         const char * __end1 = __range1 + 10L;
10        for(; __begin1 != __end1; ++__begin1)
11        {
12            const char & c = *__begin1;
13            printf("c=%c\n", static_cast<int>(c));
14        }
15
16    }
17 }
```

```
1 #include <cstdio>
2
3 int main()
4 {
5     const char arr[10]{2,4,6,8};
6
7     for(const char& c : arr)
8     {
9         printf("c=%c\n", c);
10    }
11 }
```

## Resize von struct Vektor

```
Éş xîsđ êîŋ Λêłťôş ñîť _ Éłêŋêŋťêŋ  
Éşcêuğť ũđ ñîť đêŋ Wêşť " . gêğüêłłť  
AŋşçĤîêşşêŋđ xîsđ đîê Dîŋêŋşîôŋ đêş  
Λêłťôşş ăüğ " . êşĥôêĥť ũđ kêđêş  
Éłêŋêŋť ñîť " . cêłêğť Dăŋŋ xîsđ  
gêřsüêğť ôç ăüç kêđêş đêş " .  
Éłêŋêŋťê đêŋ Wêşť " , ĥăť  
čôôł' ťêşťRêşîcê
```

```
Λêłťôş ŵ  
çôŋşť îŋť xêşť " ,  
çôŋşť îŋť wêłşîcê  
İŋîť ŵ xêşť , wêłşîcê  
çôŋşť îŋť ŋêxΛêłşîcê " .  
Rêşîcê ŵ ŋêxΛêłşîcê  
Rsüêğêŋ ăüğ ăłťêŋ Wêşťê  
ğôş îŋť î . î wêłşîcê î  
îğ ŵ đăťêŋ î xêşť ,  
Gsêê ŵ  
sêťfusŋ ğăł'sê
```

```
şêťΛăłüê ŵ xêşť  
ğôş îŋť î . î ŋêxΛêłşîcê î
```

```
îğ ŵ đăťêŋ î xêşť  
Gsêê ŵ  
sêťfusŋ ğăł'sê
```

```
Gsêê ŵ  
sêťfusŋ ťşüê
```

Wir haben Free vergessen  
bzw. ICH habe es vergessen

## Resize von struct Vektor

```
void Resize(int& n) {
    struct Node {
        int x;
        Node* next;
    };
    Node* head = nullptr;
    Node* cur = head;
    while (n-- > 0) {
        cur->next = new Node(x);
        cur = cur->next;
    }
}
```

## Vorlesungsplanung

Resize vom letzten Mal nochmals wiederholen

Was mir bei den Übungen auffiel: (Tests...)

Nächste Übungsaufgabe zur Abgabe vorstellen

Andere nützliche Dinge über Zeiger

Aufgabe Mit30\_40\_Belegen (60 Minuten als Einzelübung)

Andere nützliche (und weniger wichtige) Dinge über Zeiger

Mit Klassen beginnen ?