

Clicker

Bitte den Link

<https://vc2.sonia.de/b/har-zq1-o0p-dhs>
für WS 2024/25 nutzen.

Die verschiedenen Programmierparadigmen von C++

		Termin	Vorlesung	Übungen und Feedback
Vorles	Woche		Freitag ; Block 1+2	
4	4	18. Okt Fr	Werte- Ref-Semantik, Zeiger, Klassen string, vector<T>	
5	5	25. Okt Fr	Werte-, Zeigerpa	mit
6	7	06. Nov Mi	tiefe und flache Kopie: Kopierkonstruktor, LogTrace	(Zusatztermin in diesem Zeitraum)
7	7	08. Nov Fr	Operatoren; Templates	
	8	15. Nov Fr	fällt sehr, sehr wahrscheinlich aus	tiefe, flache Kopie, minimale Std-Schnittstelle; dyn Array Abgabe bis Fr 15.11
8	9	22. Nov Fr	STL, Iteratoren; Algorithmus versus Besuch)	
9	10	29. Nov Fr	lineare und assoziative Container;	callback-Extraction simple; Berechnung Raten, correction; Zwischen-Review 29.11
10	11	06. Dez Fr	Verschiebeoperatoren Klasse unique_ptr,, shared_ptr,	Unterstützung/Vorabnahme der Übungen; z.B 11.12 für Abgabe am 18.12 (Zusatztermin in diesem Zeitraum)
11	12	13. Dez Fr	Polymorphie; Lambda-Ausdrücke	finale Abgabe der Übungen; Di 18.12
12	13	20. Dez Fr	Vorbereitung Klausur	

Morgen, 19.10 ist Abgabe der ersten Aufgabe, Details in 5 Minuten

Klausur: Mi, 8.1.25, 11-12:30 Uhr
Klausureinsicht: Fr. 24.1.2025 10:00- 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

Feedback zur letzten Vorlesung

1. Das war alles viel zu schnell. Ich hätte gerne mehr Übungen gehabt.
2. Das war ganz viel Neues. Habe ich grob verstanden, aber beim Selbermachen wird es nebelig. Ich werde es mir zu Hause nochmals anschauen und glaube, dass sich der Nebel dann lichtet.
3. Viel Neues, aber ich denke ich kriege die Vektorübung/Sprechfunk nun auch zu Hause hin.
4. Das meiste kannte ich.

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 }
```

Vorlesungsplanung

Was mir bei den Übungen auffiel

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