

## Clicker

Bitte den Link

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

A	B	C	D	E	F
Termin				Vorlesung	Übungen und Feedback
Vorles				<b>Termine des Semesters</b>	
Woche					
1	1	27. Sep	Fr	Einführung, Worthäufigkeiten i Datentyp., Steueranw.	<b>Freitag, 1. und 2. Block: 08:15 bis 11:30</b>
2	2	02. Okt	Mi	Funktionen Strukturen; Test first	
3	3	11. Okt	Fr	Header-Dateien, Include-Wächter, Namespaces, Werte- und Ref-Semantik, Zeiger, Heap- und Stackspeicher	Abgabe Sa 12.10; Levenshtein Distanz für zwei Wortfolgen; Unterstützung
4	4	18. Okt	Fr	Werte- Ref-Semantik, Zeiger, Klassen string, vector<T>	
5	5	25. Okt	Fr	Werte-, Zeigerparameter etc., ADT; Klassen,	dyn C-Arrays, verschiedene Dateien, argv[] bei main; keine Bewertung, aber zusammen mit
7	7	06. Nov	Mi	tiefe und flache Kopie: Kopierkonstruktor, LogTrace	Unterstützung bei den Übungen per BBB (Zusatztermin in diesem Zeitraum)
8	7	08. Nov	Fr	Operatoren; Templates	
	8	15. Nov	Fr	fällt sehr, sehr wahrscheinlich aus	<b>Klausur: noch offen Uhr</b> <b>Klausureinsicht: Fr. 24.1.2025 xxx Uhr</b>
9	9	22. Nov	Fr	STL, Iteratoren; Algorithmus versus Methode (oder DLR- Besuch)	
10	10	29. Nov	Fr	lineare und assoziative Container;	Callsign Extraction simple; Berechnung Raten, correction; <b>Zwischen-Review 29.11</b>
11	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)
12	12	13. Dez	Fr	Polymorphie; Lambda-Ausdrücke	<b>finale Abgabe der Übungen; Di 18.12</b>
13	13	20. Dez	Fr	Vorbereitung Klausur	

## Termine

- Mi, 2.10 Ersatztermin für 4.11
- Mi 6.11 (16 bis 19:15 Uhr) Ersatztermin für Fr. 1.11.2024
- Fr. 15.11 fällt sehr wahrscheinlich aus
- Kurz vor den Abgaben der Übungen, individuelle Termin (ca. 20 Minuten pro Zweier- oder Einerteam) um über Stand zu sprechen (ca. 3 pro Gruppe)
- Fr. 22.11: ggf. DLR-Besuch

## Dies und Das

[Hartmut.Helmke@dlr.de](mailto:Hartmut.Helmke@dlr.de)

Bitte anmelden durch E-Mail an mich wegen SVN Zugang

Für alle die sich bei mir bis Sonntag 29.9 gemeldet haben, sollte der SVN-Zugriff eingerichtet (bis 25.9) bzw. beantragt sein.

## Rückblick auf die letzte Vorlesung

Kennenlernen

Wörterzählen in einer Datei, Frage zu Übungen dazu?

Admin-Kram

Datentypen, Kontrollstrukturen,

## Letzte Woche

1. Von der Idee zur Software
2. Funktionen und Datenstrukturen
3. Organisation des Quellcodes
4. Werte- und Referenzsemantik
5. Entwurf von Algorithmen
6. Fehlersuche und –behandlung
7. Software-Entwicklung im Team
8. Abstrakte Datentypen: Einheit von Daten und Funktionalität
9. Vielgestaltigkeit (Polymorphie)
10. Entwurfsprinzipien für Software



Anhang A: Die Familie der C-Sprachen

Anhang B: Grundlagen der C++ und der Java-Programmierung

## Clicker-“Abstimmung“

```
for (int i=0; i<3; ++i) {  
    cout << i << ", ";  
}  
for (int i=3; i>0; i--) {  
    cout << i << ", ";  
}
```

Wie ist die Ausgabe des obigen Programms auf dem Bildschirm?

1. 1, 2, 3, 3, 2, 1,
2. 1, 2, 3, 2, 1, 0
3. 0, 1, 2, 3, 3, 2, 1
4. 0, 1, 2, 3, 2, 1,

## Clicker-“Abstimmung“

```
int i = 0;
while ( i++ < 5) {
    cout << i << ", ";
}
```

Wie ist die Ausgabe des obigen Programms auf dem Bildschirm?

1. 0, 1, 2, 3, 4,
2. 1, 2, 3, 4, 5,
3. 0, 1, 2, 3,
4. 1, 2, 3, 4,

Ergebnis:

0, 1, 2, 3, 4, --  1, 2, 3, 4, 5,  0, 1, 2, 3,  1, 2, 3, 4,



## Clicker-“Abstimmung“

```
int i = 0;
while ( i < 5) {
    cout << i << ",";
    i++;
}
```

Wie ist die Ausgabe des obigen Programms auf dem Bildschirm?

1. 0, 1, 2, 3, 4,
2. 1, 2, 3, 4, 5,
3. 0, 1, 2, 3,
4. 1, 2, 3, 4,

Ergebnis:

-- 0, 1, 2, 3, 4, \_\_ 1, 2, 3, 4, 5, \_\_ 0, 1, 2, 3, \_\_ 1, 2, 3, 4,

## Clicker-“Abstimmung“

```
int main(){
    unsigned int i = 4;
    int summe = 0;
    while (i >= 0) {
        summe += i;
        --i;    cout << i << " ";
    }
    cout << " summe ist " <<summe << "\n";
}
```

Wie ist die Ausgabe des obigen Programms auf dem Bildschirm?

1. 3, 2, 1, 0, -1, summe ist 10
2. 4, 3, 2, 1, 0, summe ist 10
3. „Unendliche Ausgabe, da Todschleife“
4. 3, 2, 1, 0, summe ist 9

3, 2, 1, 0, -1, summe ist 10  4, 3, 2, 1, 0, summe ist 10  
 „Unendliche Ausgabe, da Todschleife“  3, 2, 1, 0, summe ist 9

## Nachtrag: Hexadezimale Zahlen

10 0101 bedeutet:  $1 * 2^0 + 1 * 2^2 + 1 * 2^5 = 1 + 4 + 32 = 37$

4 binäre Ziffern werden nun immer zusammengefasst, d.h.

0010 0101 wird zu 25. Wir brauchen nun 16 Ziffern.

16 Ziffern: 0,1, 2, 3 ... 9, A, B, C, D, E, F

**A9C2** bedeutet somit :  $2 * 16^0 + C * 16^1 + 9 * 16^2 + A * 16^3 =$   
 $= 2 + 12 * 16 + 9 * 256 + 10 * 4096 = 43458$

**FE** bedeutet somit :  $E * 16^0 + F * 16^1 =$   
 $= 14 + 15 * 16 = 254$

Wird verwendet, um Speicheradressen anzugeben.

## Clicker-“Abstimmung“

Welcher dezimalen Zahl entspricht hexadezimal **10A**?

- 1. 110
- 2. 266
- 3. 267
- 4. 4010

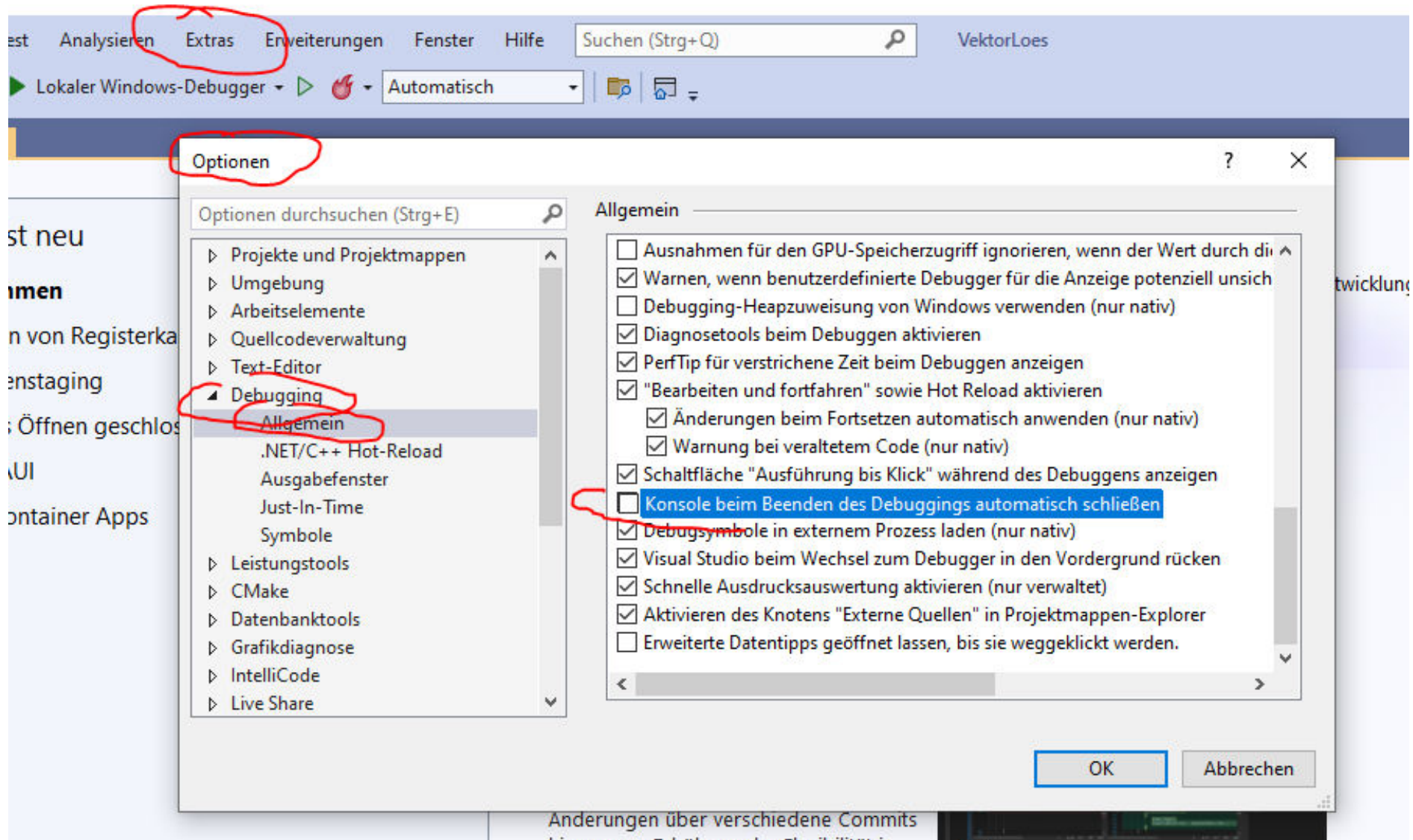
Ergebnis:  
\_\_ 110 -- 266 \_\_ 267 \_\_ 4010

Welcher dezimalen Zahl entspricht hexadezimal **B9**?

- 1. 20
- 2. 169
- 3. 185
- 4. 199

Ergebnis:  
\_\_ 20 \_\_ 169 -- 185 \_\_ 199

# Extras → Optionen → Debugging → Allgemein →



## Vorlesungsplanung

Wiederholung auch zu Hex-Zahlen: (15 Minuten)

1. Übungsaufgabe mit Abnahmekriterien vorstellen (20 Minuten)

Funktionen etc. (55 Minuten)

Pause

Übungen zu MinMaxDurch (30 Minuten)

Structs und Records (30 Minuten)

Gemeinsame Übungen zu Vektor als Struktur (30 Minuten)