

Probeklausur


Grundlagen der Programmierung

Wintersemester 2025/26 | HTW Berlin

Prof. Dr. Alexandra Mikityuk

Prüfungsinformationen

- **Bearbeitungszeit:** 90 Minuten
- **Gesamtpunktzahl:** 100 Punkte
- **Hilfsmittel erlaubt:**
 - Vorlesungsunterlagen (gedruckt oder digital)
 - Eigene Notizen
 - Laptop, Tablet oder andere technische Geräte
- **Hilfsmittel NICHT erlaubt:**
 - Keine KI-Tools (ChatGPT, Claude, Copilot, etc.)
 - Keine Google-Suche oder andere Suchmaschinen
 - Keine Kommunikation mit anderen Personen
- **Hinweis:** Schreiben Sie leserlich und strukturiert. Unlesbare Antworten können nicht bewertet werden.
- **Code-Aufgaben:** Geben Sie vollständigen, lauffähigen Code an (inkl. `#include`, `main()`, etc.)

 **Wichtig:** Bitte tragen Sie auf jeder Seite Ihren Namen und Ihre Matrikelnummer ein!

Teil 1: Theorie und Verständnis (30 Punkte)

Aufgabe 1.1: Algorithmen und Programmierung

8 Punkte

a) Erklären Sie den Unterschied zwischen einem **Algorithmus** und einem **Programm**. Geben Sie je ein Beispiel. (4 Punkte)

b) Was versteht man unter **deterministisch** im Kontext von Algorithmen? Warum ist diese Eigenschaft wichtig? (4 Punkte)

Aufgabe 1.2: Datentypen und Format-Specifier**10 Punkte**

a) Vervollständigen Sie die folgende Tabelle mit den korrekten Format-Specifiern für `printf()` und `scanf()`: (6 Punkte)

Datentyp	<code>printf()</code> Format	<code>scanf()</code> Format	Beispiel-Verwendung
int	_____	_____	Ganze Zahlen (Alter, Anzahl)
double	_____	_____	Kommazahlen (Preis, Temperatur)
char	_____	_____	Einzelne Zeichen (j/n, +/-)

b) Erklären Sie den Unterschied zwischen `int` und `double`. Wann verwendet man welchen Datentyp? (4 Punkte)

Aufgabe 1.3: Operatoren und Ausdrücke**12 Punkte**

Gegeben seien folgende Variablendeklarationen:

```
int a = 10, b = 3;  
double x = 7.5, y = 2.0;
```

Berechnen Sie das Ergebnis der folgenden Ausdrücke. Geben Sie auch den **Datentyp** des Ergebnisses an:

Ausdruck	Ergebnis	Datentyp
$a + b$	_____	_____
$a - b$	_____	_____
$a * b$	_____	_____
a / b	_____	_____
$a \% b$	_____	_____
x / y	_____	_____

Teil 2: Code-Analyse und Fehlersuche (25 Punkte)

Aufgabe 2.1: Code-Analyse**8 Punkte**

Analysieren Sie den folgenden Code:

```
#include <stdio.h>

int main()
{
    int n = 5;
    int ergebnis = 1;

    for(int i = 1; i <= n; i++)
    {
        ergebnis = ergebnis * i;
    }

    printf("Ergebnis: %d\n", ergebnis);
    return 0;
}
```

a) Was ist die Ausgabe dieses Programms für $n = 5$? Zeigen Sie die Werte von `ergebnis` nach jedem Schleifendurchlauf: (4 Punkte)

b) Beschreiben Sie in eigenen Worten, was dieses Programm macht. (2 Punkte)

c) Was ist die Ausgabe, wenn $n = 0$? Erklären Sie, warum. (2 Punkte)

Aufgabe 2.2: Fehlersuche (Debugging)**12 Punkte**

Der folgende Code soll die Durchschnittsnote von drei Klausuren berechnen, enthält aber mehrere Fehler. Finden Sie **alle Fehler** und erklären Sie, warum sie falsch sind:

```
#include <studio.h>

int main()
{
    int note1 = 1.3;
    int note2 = 2.0;
    int note3 = 1.7;

    int durchschnitt = (note1 + note2 + note3) / 3;

    printf("Durchschnittsnote: %f\n", durchschnitt);

    return 0
}
```

Liste der Fehler (mindestens 5):

Aufgabe 2.3: Code-Trace**5 Punkte**

Verfolgen Sie den folgenden Code Schritt für Schritt und geben Sie die Ausgabe an:

```
#include <stdio.h>

int main()
{
    int x = 5;
    int y = 10;

    if(x > 3)
    {
        y = y + x;
    }
    else
    {
        y = y - x;
    }

    if(y > 12)
    {
        printf("A: %d\n", y);
    }
    else if(y > 10)
    {
        printf("B: %d\n", y);
    }
    else
    {
        printf("C: %d\n", y);
    }

    return 0;
}
```

Ausgabe:

Erklärung (zeigen Sie die Werte von x und y nach jedem Schritt):



Teil 3: Praktische Programmierung (45 Punkte)

Aufgabe 3.1: Temperaturumrechnung

12 Punkte

Schreiben Sie ein **vollständiges C-Programm**, das:

- Eine Temperatur in Celsius vom Benutzer einliest
- Diese Temperatur in Fahrenheit umrechnet (Formel: $F = C \times 1.8 + 32$)
- Beide Temperaturen mit **einer Nachkommastelle** ausgibt
- Eine sinnvolle Meldung ausgibt (z.B. "20.0°C entspricht 68.0°F")

***Hinweis:** Achten Sie auf den richtigen Datentyp für Kommazahlen!*

Aufgabe 3.2: Notensystem mit Verzweigungen

15 Punkte

Schreiben Sie ein **vollständiges C-Programm**, das:

- Eine Punktzahl (0-100) vom Benutzer einliest
- Die entsprechende Note ausgibt nach folgendem Schema:
 - 90-100 Punkte: "Sehr gut (1.0)"
 - 80-89 Punkte: "Gut (2.0)"
 - 70-79 Punkte: "Befriedigend (3.0)"
 - 60-69 Punkte: "Ausreichend (4.0)"
 - 0-59 Punkte: "Nicht bestanden (5.0)"
- Bei ungültiger Eingabe (< 0 oder > 100) eine Fehlermeldung ausgibt

Tipp: Verwenden Sie *if-else if-else* Strukturen. Denken Sie an die Reihenfolge der Bedingungen!

Aufgabe 3.3: Einfacher Rechner


18 Punkte

Schreiben Sie ein **vollständiges C-Programm** für einen einfachen Taschenrechner, der:

- Zwei Zahlen (Kommazahlen!) vom Benutzer einliest
- Einen Operator (+, -, *, /) als Zeichen einliest
- Die entsprechende Berechnung durchführt
- Das Ergebnis mit **zwei Nachkommastellen** ausgibt
- **Besondere Fälle behandelt:**
 - Division durch Null: Fehlermeldung ausgeben!
 - Ungültiger Operator: Fehlermeldung ausgeben!

Beispiel-Ausgabe:*Erste Zahl: 10.5**Zweite Zahl: 2.0**Operator (+, -, *, /): ***Ergebnis: 10.50 * 2.00 = 21.00*

Gesamtpunktzahl: _____ / 100 Punkte

Viel Erfolg! 

Vergessen Sie nicht, Ihre Lösungen zu überprüfen und auf Vollständigkeit zu achten.