

Übungsklausur Informatik II

für Studierende des Fachbereiches MK

Name, Vorname	Matrikelnummer
Punkte	Note

Bearbeitungszeit: 90 Minuten

Hilfsmittel: Skript, Mitschriften, Bücher, Taschenrechner

Sehr geehrte Studierende,
bitte

- tragen Sie Ihren Namen und Ihre Matrikelnummer ein
- schalten Sie Ihr Handy aus (Angeschaltete Handy gelten als Täuschungsversuch.)
- legen Sie Ihren Studien- und Lichtbildausweis zur Kontrolle bereit
- schreiben Sie nur auf das ausgeteilte Klausurformular und lassen Sie dieses bitte geheftet (Weitere Schmier- oder Notizzettel sind nicht erlaubt.)
- beachten Sie, dass jede Zuwiderhandlung und jeder Täuschungsversuch zum sofortigen Ausschluss von der Prüfung führt

Eine Teilnahme an der Klausur setzt die erfolgreiche Teilnahme am Praktikum voraus.

Die maximale nötige Punktzahl (=100%) beträgt 100 Punkte.

Wichtig: Nur bei ersichtlicher Herleitung des Ergebnisses ist die Vergabe von Teilpunkten möglich.

Viel Erfolg !

Aufgabe 1:

Welche Werte können Variablen speichern, wenn sie als

1. 'int' deklariert sind?

...

(je 1 Punkt)

Antwort zu 1:

Aufgabe 2:

Bewerten Sie folgende Behauptungen zu Konstruktor / Destruktor mit ja oder nein:

(je 1 Punkt)

1. Der Destruktor dient zur Initialisierung der Eigenschaften einer Klasse.
2. Der Standardkonstruktor besitzt keine Parameter.

...

Antwort zu 1:

Antwort zu 2:

Aufgabe 3:

Bewerten Sie folgende Behauptungen zu Klassen / Objekte mit ja oder nein:

(je 1 Punkt)

1. Ist kein Zugriffsrecht angegeben, gilt automatisch das Schlüsselwort public.
2. Das Schlüsselwort private erlaubt den Zugriff von außerhalb der Klasse.

Antwort zu 1:

Antwort zu 2:

...

Aufgabe 4:

Was gibt folgendes Programm aus?
(8 Punkte)

```

enum Tier {Elefant, Hase, Schildkröte, Tiger, Känguruh};
enum Eigenschaft {langsam, normal, schnell};

void main()
{
    Tier meinTier;
    meinTier = Tiger;

    for(int geschwdkt=langsam; geschwdkt<=schnell; geschwdkt++)
    {
        if(geschwdkt == schnell)
        {
            cout << "Tier: " << meinTier << endl;
            cout << "Geschwindigkeit: " << geschwdkt << endl;
        }
    }
}

```

Antwort: Tier : 3
Geschwindigkeit : 2

Aufgabe 5:

Welchen Wert gibt dieses Programm aus?
(10 Punkte)

```

global → int zahl = 4;
void main()
{
    lokal → int zahl = 3;
    {
        → int zahl = 2;
        cout << zahl + ::zahl;
    }
}

```

↑ ↑
2 4

Antwort: 6

Aufgabe 6:

Welcher Wert wird ausgegeben?
(10 Punkte)

```
#define TESTWERT 1
#define TESTWERT2 2
#define TESTWERT3 3

void main()
{
    #define AUSGABE
    #ifndef AUSGABE
    cout << TESTWERT << endl; )
    #elif (TESTWERT2 == 2)
    cout << TESTWERT2 << endl;
    #else
    cout << TESTWERT3 << endl;
    #endif
}
```

Antwort:

2

Aufgabe 7:

Welchen Wert gibt folgendes Programm aus?
(15 Punkte)

```
int Addition(int parameter)
{
    return (parameter + 4);
}

void Berechnung(int konstante, int &zahl)
{
    zahl = Addition(konstante);
}

void main()
{
    const int meineKonstante = 4;
    int zahl = 2;

    Berechnung(meineKonstante, zahl);
    Addition(zahl);
    cout << zahl + Addition(3);
}
```

aber ist $zahl = 8$
 \Rightarrow Ergebnis = 12, aber irrelevant, da nicht gespeichert

Antwort:

15

Aufgabe 8:

Welchen Wert gibt dieses Programm aus?
(10 Punkte)

$$\begin{aligned} \text{feld}[0] &= \text{feld}[0] + \text{feld}[0] = 2 + 2 = 4, \\ [0] &= [0] \quad [1] = 4 + 3 = 7 \\ [0] &= [0] + [2] = 7 + 4 = 11 \end{aligned}$$

```
void Feldberechnung(int feld[])
{
    for(int i=0; i<3; i++)
        feld[0] = feld[0] + feld[i];
}
```

```
void main()
{
    int feld[3] = {1, 2, 3};
    for(int i=0; i<3; i++)
        feld[i] += 1;
    Feldberechnung(feld);
    cout << feld[0]+feld[1]+feld[2];
}
```

$$\begin{aligned} \Rightarrow \text{feld}[0] &= \text{feld}[0] + 1 = 1 + 1 = 2 \\ [1] &= 2 + 1 = 3 \\ [2] &= 3 + 1 = 4 \\ \text{feld} &= \{2, 3, 4\} \end{aligned}$$

Antwort:

18

Aufgabe 9:

Was gibt folgendes Programm aus?
(20 Punkte)

```

class CKlasse1
{
private:
    int m_i, m_j;
public:
    CKlasse1()
    {
        m_i = 1; m_j = 2;
    }
    int Berechne(int f)
    {
        return m_i + m_j + f;
    }
    void Ausgabe()
    {
        cout << "Das Ergebnis lautet: ";
    }
};

class CKlasse2 : public CKlasse1
{
private:
    int m_k;
public:
    CKlasse2(int zahl):m_k(zahl)
    {
    }
    int Berechne(int f)
    {
        return CKlasse1::Berechne(f) + m_k + f;
    }
    void Ausgabe()
    {
        cout << "Die Zahl lautet: ";
    }
};

void main()
{
    CKlasse2 Objekt1(2);
    CKlasse1 Objekt2;

    Objekt2.Ausgabe();
    cout << Objekt2.Berechne(2) + Objekt1.Berechne(3) << endl;
}

```

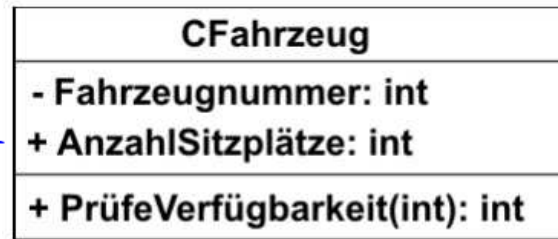
$m_i = 1$
 $m_j = 2$

$m_k = 2$

Antwort: Das Ergebnis lautet: 16

Aufgabe 10:

Setzen Sie das gegebene UML-Diagramm in die Klassenstruktur um. Die angegebenen Funktionen **nicht** ausprogrammieren. (Der Funktionsrumpf {} in der .cpp-Datei bleib leer.)
(15 Punkte)



} Name der Klasse
} Variablen
} Funktionen



}
}

Fahrzeug.h

```

class CFahrzeug
{
private: int Fahrzeugnummer;
public: int AnzahlSitzplaetze;
        int PruefeVerfuegbarkeit(int);
};
  
```

Kraftfahrzeug.h

```

#include "Fahrzeug.h"
class CKraftfahrzeug : public CFahrzeug
{
private: int Hochstgeschw.;
protected: int PruefeF. ();
public: int GetGeschw ();
        void SetGeschw (int);
};
  
```

Fahrzeug.cpp

```

#include "Fahrzeug.h"
int CFahrzeug::PruefeVerfuegbarkeit(int a)
{ }
  
```

Kraftfahrzeug.cpp

```

#include "Kraftfahrzeug.h"
int CKraftfahrzeug::PruefeFahrerlaubnis()
{ }
int " " :: GetGeschw. ()
{ }
void " " :: SetGeschw (int a)
{ }
  
```