

Name, Vorname:

Legi-Nummer:

Ich bestätige mit meiner Unterschrift, dass ich diese Prüfung unter regulären Bedingungen ablegen konnte, und dass ich die allgemeinen Richtlinien gelesen und verstanden habe.

I confirm with my signature that I was able to take this exam under regular conditions and that I have read and understood the general guidelines.

Unterschrift:

Allgemeine Richtlinien:

General guidelines:

1. Dauer der Prüfung: 60 Minuten.
2. Erlaubte Unterlagen: Wörterbuch (für gesprochene Sprachen).
3. Benützen Sie einen Kugelschreiber (blau oder schwarz) und keinen Bleistift. Bitte schreiben Sie leserlich. Nur lesbare Resultate werden bewertet.
4. Lösungen sind direkt auf das Aufgabenblatt in die dafür vorgesehenen Boxen zu schreiben (und direkt darunter, falls mehr Platz benötigt wird). Ungültige Lösungen bitte deutlich durchstreichen! Korrekturen bei Multiple-Choice Aufgaben unmissverständlich anbringen!
5. Falls Sie sich durch irgendjemanden oder irgendetwas gestört fühlen, melden Sie dies sofort der Aufsichtsperson.
6. Wir sammeln die Prüfung zum Schluss ein. Wichtig: stellen Sie unbedingt selbst sicher, dass Ihre Prüfung von einem Assistenten eingezogen wird. Stecken Sie keine Prüfung ein und lassen Sie Ihre Prüfung nicht einfach am Platz liegen. Dasselbe gilt, wenn Sie früher abgeben wollen: bitte melden Sie sich lautlos, und wir holen die Prüfung ab. Vorzeitige Abgaben sind nur bis 15 Minuten vor Prüfungsende möglich.
7. Wenn Sie zur Toilette müssen, melden Sie dies einer Aufsichtsperson durch Handzeichen. Es darf zur gleichen Zeit immer nur eine Studentin oder ein Student zur Toilette.
8. Wir beantworten keine inhaltlichen Fragen während der Prüfung. Kommentare zur Aufgabe schreiben Sie bitte auf das Aufgabenblatt.

- Exam duration: 60 minutes.*
- Permitted examination aids: dictionary (for spoken languages).*
- Use a pen (black or blue), not a pencil. Please write legibly. We will only correct solutions that we can read.*
- All solutions must be written directly onto the exercise sheets in the provided boxes (and directly below, if more space is needed). Invalid solutions need to be crossed out clearly. Corrections to answers of multiple choice questions must be provided without any ambiguity.*
- If you feel disturbed by anyone or anything, immediately let the supervisor of the exam know this.*
- We collect the exams at the end. Important: you must ensure that your exam has been collected by an assistant. Do not take any exam with you and do not leave your exam behind on your desk. The same applies when you want to finish early: please contact us silently and we will collect the exam. Handing in your exam preliminarily is only possible until 15 minutes before the exam ends.*
- If you need to go to the toilet, raise your hand and wait for a supervisor. Only one student can go to the toilet at a time.*
- We will not answer any content-related questions during the exam. Please write comments referring to the tasks on the exam sheets.*

Aufgabe	1	2	3	4	5	6	7	8	9	Σ
Punkte										
Maximum	6	6	8	4	6	6	6	6	6	54

1 Typen und Werte 1 (Basistypen) (6 Punkte)

Geben Sie für jeden der drei Ausdrücke auf der rechten Seite jeweils C++-Typ und Wert an!

Wir merken an, dass z.B. `integer` oder `boolean` keine C++-Typen sind und als falsche Antworten gewertet werden. Die korrekten Typen heissen `int` und `bool`.

Variablen `u`, `i` und `d` seien deklariert und initialisiert wie folgt.

```
unsigned int u = 5;
```

```
int i = -10;
```

```
float f = 0.25f;
```

For each of the 3 expressions on the right, provide the C++ type and value!

Note that, for example, `integer` or `boolean` are not C++ types and will be considered as incorrect answers. The correct types are called `int` and `bool`.

Variables `u`, `i` and `d` have been initialized as shown above.

(a) `f * 0.5 * 0.32e2 == u / 2 * 2`

1 P

Typ/Type: `bool`

(b) `f * 0.5 * 0.32e2 == u / 2 * 2`

1 P

Wert/Value: `true`

(c) `u * -i * f`

1 P

Typ/Type: `float`

(d) `u * -i * f`

1 P

Wert/Value: `12.5`

(e) `++u - ++i + 10`

1 P

Typ/Type: `unsigned int`

(f) `++u - ++i + 10`

1 P

Wert/Value: `25`

2 Typen und Werte 2 (Arrays und Structs) (6 Punkte)

Geben Sie für jeden der drei Ausdrücke auf der rechten Seite jeweils C++-Typ und Wert an!

Wir merken an, dass z.B. `integer` oder `boolean` keine C++-Typen sind und als falsche Antworten gewertet werden.

Die Variablen `x` und `y` seien deklariert und initialisiert wie folgt.

```
struct my_struct {  
    int a;  
    double b;  
    int c;  
};
```

```
my_struct x = { 1, 1.5, 3 };  
int y[] = { 1, 1.5, 3 };
```

For each of the 3 expressions on the right, provide the C++ type and value!

Note that, for example, `integer` or `boolean` are not C++ types and will be considered as incorrect answers.

Variables `x` and `y` have been declared and initialized as shown above.

(a) `&y[0] == &x.a`

1 P

Typ/Type: `bool`

(b) `&y[0] == &x.a`

1 P

Wert/Value: `false`

(c) `(y[2] / y[x.a]) - (x.c / x.b)`

1 P

Typ/Type: `double`

(d) `(y[2] / y[x.a]) - (x.c / x.b)`

1 P

Wert/Value: `1`

(e) `*(y + y[2] - x.c) * x.c / x.b`

1 P

Typ/Type: `double`

(f) `*(y + y[2] - x.c) * x.c / x.b`

1 P

Wert/Value: `2`

3 Programmausgaben (8 Punkte)

Betrachten Sie folgende Funktion. Beantworten Sie die Fragen auf der rechten Seite!

```
// PRE: b > 1
// POST:
unsigned int f (unsigned int i, unsigned int b)
{
    if (i == 0) return 0;
    return 1 + f (i/b, b);
}
```

Consider the function above. Answer the questions on the right-hand side!

-
- (a) Was ist die Nachbedingung der Funktion `f`, die auf der linken Seite definiert wurde? 2 P
What is the postcondition of the function `f` defined on the left-hand side?

```
returns the number of digits in the base-b representation of i if  
i > 0, and returns 0 if i == 0
```

-
- (b) Was gibt das Programm mit folgender `main`-Funktion aus? 2 P
What is the output of the program with the following `main`-function?

```
int main() {  
    std::cout << f (10, 2);  
    return 0;  
}
```

```
4
```

-
- (c) Was gibt das Programm mit folgender `main`-Funktion aus? 2 P
What is the output of the program with the following `main`-function?

```
int main() {  
    std::cout << f ( f (65536, 2), 2);  
    return 0;  
}
```

```
5
```

-
- (d) Was gibt das Programm mit folgender `main`-Funktion aus? 2 P
What is the output of the program with the following `main`-function?

```
int main() {  
    int i = 0;  
    while ( f (++i, 3) <= 2)  
        std::cout << i << " ";  
    return 0;  
}
```

```
1 2 3 4 5 6 7 8
```

4 BNF I (4 Punkte)

Die folgende BNF definiert eine Sprache zur Beschreibung von Mengen. Beantworten Sie die Fragen auf der rechten Seite! **Anmerkung:** Leerschläge sind im Rahmen der BNF bedeutungslos. $\{1, \{2\}\}$ und $\{1,\{2\}\}$ sind also im Sinne dieser Aufgabe die gleichen Zeichenketten.

```
set = "{" elements "}"
elements = items | sets
items = item | item "," items
item = "1" | "2" | "3" | "4"
sets = set | set "," sets
```

The BNF above defines a language for the description of sets. Answer the questions on the right side! **Remark:** Whitespaces are irrelevant in the context of BNF. $\{1, \{2\}\}$ and $\{1,\{2\}\}$ are therefore to be considered equal for the purposes of this assignment.

(a) Wahr oder falsch? *true or false?*

false

1 P

Die folgende Zeichenkette ist gültig nach der BNF:

The following string is valid according to the BNF:

{1, {2}}

(b) Wahr oder falsch? *true or false?*

true

1 P

Die folgende Zeichenkette ist gültig nach der BNF:

The following string is valid according to the BNF:

{{{{1}}}}

(c) Wahr oder falsch? *true or false?*

true

1 P

Die folgende Zeichenkette ist gültig nach der BNF:

The following string is valid according to the BNF:

{{1, 2}, {{{3}}}}

(d) Wahr oder falsch? *true or false?*

true

1 P

Die folgende Zeichenkette ist gültig nach der BNF:

The following string is valid according to the BNF:

{{1, 2}, {3, 4}}

5 BNF II (6 Punkte)

Wir betrachten wieder die BNF aus der vorherigen Aufgabe. Beantworten Sie die Fragen auf der rechten Seite!

```
set = "{" elements "}"
elements = items | sets
items = item | item "," items
item = "1" | "2" | "3" | "4"
sets = set | set "," sets
```

We again consider the BNF from the previous assignment. Answer the questions on the right-hand side!

-
- (a) Wie viele verschiedene Terminalsymbole hat die BNF? 1 P
How many distinct terminal symbols does the BNF have?

A: 7

-
- (b) Wie viele verschiedene Nichtterminalsymbole hat die BNF? 1 P
How many distinct nonterminal symbols does the BNF have?

B: 5

-
- (c) Wie muss die BNF erweitert werden, so dass sie zusätzlich auch leere Mengen (`{}`) erlaubt? 1 P
Geben Sie eine passende Regel für das Symbol `set` an!
How do you have to enhance the BNF such that it in addition also allows empty sets (`{}`)?
Provide a matching rule for the symbol `set`!

C: `set = "{" elements "}" | "{}"`

-
- (d) Wie muss die BNF verändert werden, damit Mengen als Elemente alle zweistelligen Zahlen enthalten können? Wir erlauben nur strikt zweistellige Zahlen, inklusive 00,01,...,09. Geben Sie passende Regeln für die Symbole `item` und `neu digit` an! 3 P
How do you have to change the BNF such that sets may contain arbitrary two-digit numbers as elements? We only allow strictly two-digit numbers, including 00,01,...,09. Provide matching rules for the two symbols `item` and `digit`.

D: `item = digit digit`
`digit = "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"`

6 Arithmetic Stack (6 Punkte)

Folgender Code implementiert einen Teil eines sogenannten arithmetischen Stacks. Beantworten Sie die Fragen auf der rechten Seite.

```
#include <iostream>
class arithmetic_stack{
    double stack[8]; // stack values
    int position;
public:
    arithmetic_stack() {
        position = 0;
    }
    // PRE: position < 8; post: pushes value to new stack top
    void push(double value) {
        stack[position++] = value;
    }
    // PRE: position > 0; post: pops and returns value at stack top
    double pop() {
        return stack[--position];
    }
    // PRE: position > 1; post: replaces two top-most stack values by the sum
    void add() {
        push(pop() + pop());
    }
    // PRE: position > 1; post: replaces two top-most stack values by the product
    void mul() {
        push(pop() * pop());
    }
}
int main() {
    arithmetic_stack stack;
    stack.push(30); stack.push(20); stack.push(10);
    stack.add(); stack.mul();
    std::cout << stack.pop();
    return 0;
}
```

The code above implements a part of a so called arithmetic stack. Answer the questions on the right-hand side.

(a) Was gibt die Mainfunktion aus?

1 P

What is the output of the main function?

Antwort / answer:

(b) Angenommen, a, b und c seien Variablen vom Typ double. Streichen Sie in den folgenden Codefragmenten zu entfernende Anweisungen in grauen Boxen durch, so dass der angegebene Ausdruck berechnet wird.

2 P

Assume a, b and c are variables of type double. In the following code fragments, strike through statements to be removed in grey boxes such that the provided expressions are computed.

```
// compute a * b + c
arithmetic_stack s;
s.push(a); s.push(b);
 
s.push(c);
 
std::cout << s.pop();
```

```
// compute a + b * c
arithmetic_stack s;
s.push(a); s.push(b);
 
s.push(c);
 
std::cout << s.pop();
```

(c) Vervollständigen Sie folgende Funktion so, dass sie das Skalarprodukt $\sum_{i=0}^{len-1} a_i \cdot b_i$ von $a \in \mathbb{R}^{len}$ und $b \in \mathbb{R}^{len}$ zurückgibt.

3 P

Complete the following function such that it returns the scalar product $\sum_{i=0}^{len-1} a_i \cdot b_i$ of $a \in \mathbb{R}^{len}$ and $b \in \mathbb{R}^{len}$.

```
double scalarproduct(double a[], double b[], int len) {
    arithmetic_stack stack;
    ;
    for (int i = 0; i < len; ++i){
        stack.push(a[i]); stack.push(b[i]);
        ;
        ;
    }
    return stack.pop();
}
```

7 Prüfsummenberechnung (6 Punkte)

Eine Prüfsumme wird aus einer Reihe von Werten gebildet und wird verwendet um deren Integrität zu überprüfen. Vervollständigen Sie folgende Klasse, welche die Funktionalität zur Berechnung einer sehr einfachen Prüfsumme natürlicher Zahlen kapselt.

1. Konstruktor Checksum: initialisiert `sum` mit 0.
2. Member-Funktion `add`: Addiert eine nichtnegative ganze Zahl zu `sum`.
3. Member-Funktion `get`: Gibt den Wert `sum` der berechneten Prüfsumme zurück. `sum` wird nicht geändert.

```
class Checksum {
    D1 unsigned int sum;
    A :
    Checksum() : B {}

    void add(D2 unsigned int value) D3 {
        C ;
    }
    unsigned int get() D4 {
        return E ;
    }
};
```

A checksum is generated out of a series of values and is used to verify the integrity of these values. Complete the class `Checksum` that implements functionality to calculate a very simple checksum of natural numbers.

1. Constructor `Checksum`: initializes member variable `sum` with 0.
2. Member function `add`: Adds an unsigned int value to the member variable `sum`.
3. Member function `get`: Computes and returns the checksum `sum`. It does not modify `sum`.

-
- (a) Welche Deklaration muss bei A eingesetzt werden, so dass folgendes Programm kompiliert?
Fill in the declaration in A, to make the following program compile. 1 P

```
#include <iostream>
int main() {
    Checksum cs;
    cs.add (3); cs.add (4); cs.add (5);
    std::cout << cs.get() << "\n"; // 12
}
```

A: `public`

-
- (b) Welche Initialisierungsanweisung muss bei B eingesetzt werden?
Fill in the initialisation statement for B. 1 P

B: `sum(0)`

-
- (c) Welche Anweisung muss bei C eingesetzt werden?
Fill in the statement for C. 1 P

C: `sum += value`

-
- (d) Betrachten Sie Markierungen D1, D2, D3 und D4. Nennen Sie diejenigen Markierungen, an denen das Schlüsselwort `const` eingesetzt werden kann, so dass die Klasse noch kompiliert werden kann.
Look at the markings D1,D2,D3,D4. Name those markings at which the keyword `const` can be inserted such that the class can still be compiled. 2 P

D: `D2 D4`

-
- (e) Welcher Ausdruck muss für E eingesetzt werden?
Which expression must be inserted for E? 1 P

E: `sum`

8 Fließkommazahlen (6 Punkte)

Für die Fragen rechts, verwenden Sie den IEEE-Standard für Fließkommazahlen.

For the questions on the right, assume the IEEE standard for floating point arithmetics.

(a) Wahr oder falsch? *true or false?* 1 P

Das Literal `1e7f` bezeichnet denselben Wert wie das Literal `8e4f`.
The literal `1e7f` has the same value as the literal `8e4f`.

(b) Wahr oder falsch? *true or false?* 1 P

Für zwei Fließkommavariablen `a` und `b` mit beliebigem Wert gilt:
For any two floating point variables `a` and `b`, we have:

`a * b == b * a`

(c) Wahr oder falsch? *true or false?* 1 P

Für drei Fließkommavariablen `a`, `b`, und `c` mit beliebigem Wert gilt:
For any three floating point variables `a`, `b`, and `c`, we have:

`a + b + c == c + b + a`

(d) Wahr oder falsch? *true or false?* 1 P

Für eine Fließkommavariablen `a` mit beliebigem Wert gilt:
For any floating point variable `a`, we have:

`a + 1/2 == a + 0.5`

(e) Wahr oder falsch? *true or false?* 1 P

Der Typ `double` hat genau zwei mal so viele signifikante Bits wie der Typ `float`.
The type `double` has exactly twice as many significant bits as `float`.

(f) Wahr oder falsch? *true or false?* 1 P

Für `a > 0` vom Typ `double` ist `std::sqrt(a)` die exakte Wurzel von `a`.
For `a > 0` of type `double`, `std::sqrt(a)` is the exact square root of `a`.

9 Referenzen (6 Punkte)

Betrachten Sie das folgende Programm, das eine Funktion `foo` aufruft. Beantworten Sie die Fragen auf der rechten Seite für drei verschiedene Varianten der Funktion `foo`!

```
int main()
{
    int a = 0;
    int b = 1;
    for (int i=0; i<5; ++i) {
        b = foo (a, b);
        std::cout << b << " ";
    }
    return 0;
}
```

Consider the program above that calls a function `foo`. Answer the questions on the right for three different variants of the function `foo`!

-
- (a) Was ist die Ausgabe des Programms für die folgende Variante von foo?
What is the output of the program for the following variant of foo?

2 P

```
int foo (int& a, int b)
{
    a += b;
    return a;
}
```

A: 1 2 4 8 16

-
- (b) Was ist die Ausgabe des Programms für die folgende Variante von foo?
What is the output of the program for the following variant of foo?

2 P

```
int foo (int a, int b)
{
    a += b;
    return a;
}
```

B: 1 1 1 1 1

-
- (c) Was ist die Ausgabe des Programms für die folgende Variante von foo?
What is the output of the program for the following variant of foo?

2 P

```
int foo (int a, int& b)
{
    a += b;
    return a;
}
```

C: 1 1 1 1 1