

Aufgabe 1 (10 Punkte)

- a) Finden Sie genau 3 Fehler in folgendem Programm, die dazu führen, dass der Quellcode nicht kompiliert wird (Folgefehler zählen nicht mit). Markieren Sie die Stellen und beschreiben Sie kurz den Fehler.

(6 Punkte)

```
int[] a = { 2,3,5,7,11,13 };  
int* p1 = &a[0], p2 = &a[1];  
for (int i=0, j=i; i<sizeof(a)/sizeof(int); ++i, ++j)  
    std::cout << "i=" << a(i) << std::endl;
```

Fehlerbeschreibung:

- b) Finden Sie heraus, an welcher Stelle folgender Code zuerst mit einem Fehler abbricht. Markieren Sie die Stelle und geben Sie einen Korrekturvorschlag an.

(4 Punkte)

Funktion myggt:

```
int myggt(int a, int b, int ggt) {  
    int n=0;  
    for (; b!=0; ++n) { int h=a%b; a=b; b=h; }  
    ggt = a;  
    return n;  
}
```

Aufruf:

```
int a=6,b=9,ggt=0;  
int loops = myggt(a,b,ggt);  
  
std::cout << "bruch=" << a << "/" << b << std::endl;  
std::cout << "bruch=" << a/ggt << "/" << b/ggt << std::endl;  
std::cout << "loops=" << loops << std::endl;
```

Korrekturvorschlag:

Aufgabe 2 (12 Punkte)

a) Geben Sie an, was durch folgenden Code ausgegeben wird.

(6 Punkte)

```
const int n = 4;
int p[] = {2,3,5,7};

for (int i=1; i<n; ++i)
    std::cout << p[i] << " ";
for (int* pp=p; pp<p+n; ++pp)
    std::cout << (*(pp++)) << " ";
```

Ausgabe (Ziffernfolge):

b) Folgende Klassen sind gegeben.

```
struct A2_Base {
    void f1() { std::cout << "1 "; }
    virtual void f2() { std::cout << "2 "; }
};

struct A2_ErbErb : public A2_Base {
    void f1() { A2_Base::f2(); }
    void f2() { std::cout << "4 "; }
};
```

Geben Sie an, was durch folgenden Code ausgegeben wird.

(6 Punkte)

```
A2_Base A;      A.f1();  A.f2();
A2_ErbErb B;    B.f1();  B.f2();
A2_Base* C = &B; C->f1(); C->f2();
```

Ausgabe (Ziffernfolge):

Aufgabe 3 (14 Punkte)

- a) Deklarieren und initialisieren Sie einen Zeiger auf einen Integer `n`.

(2 Punkte)

- b) Deklarieren und initialisieren Sie eine Referenz auf einen Integer `n`.

(2 Punkte)

- c) Geben Sie eine Struktur `Punkt` an, die genau zwei Member-Variablen `x`, `y` vom Typ Integer enthält.

(4 Punkte)

- d) Geben sie eine Klasse `CPunkt` an, die zwei Member-Variablen `x`, `y` vom Typ Integer und einen Konstruktor ohne Argumente enthält. Der Konstruktor soll `x` und `y` mit 0 in der Initialisierungsliste initialisieren. Jeder Member der Klasse soll für jeden, der die Klasse verwenden möchte, zugänglich sein.

(6 Punkte)

Aufgabe 4 (10 Punkte)

Entscheiden Sie bitte je Teilaufgabe, welcher der folgenden Sätze (1) bis (4) korrekt ist (genau eine richtige Antwort).

a) Ihre Antwort: ____

- (1) &A liefert immer die Adresse der Variablen A.
- (2) &A liefert üblicherweise die Adresse der Variablen A, kann aber für eigene Klassen selbst definiert werden.
- (3) &A dereferenziert die Variable A.
- (4) &A ist ein ungültiger Ausdruck und führt zu einem Syntaxfehler.

b) Ihre Antwort: ____

- (1) Eine Klasse muss immer in einer Header-Datei (*.h) definiert sein.
- (2) Eine Klasse muss zum Zeitpunkt der Verwendung bekannt sein.
- (3) Eine Klasse kann verwendet werden, auch wenn sie erst später aber in der gleichen Quelltext-Datei (*.cpp) definiert wird.
- (4) Eine Klasse muss nur als `extern` deklariert sein und kann dann immer verwendet werden.

c) Ihre Antwort: ____

- (1) Ein C-String ist ein unveränderliches Array (const) von Zeichen (chars).
- (2) Ein C-String ist ein Array von Zeichen, welches zu Beginn die Länge des Arrays enthält.
- (3) Ein C-String ist ein Array von Zeichen, welches mit dem 0-Character endet.
- (4) Ein C-String ist die Bezeichnung für die Klasse `std::string`.

d) Ihre Antwort: ____

- (1) Es gibt keine Mehrfachvererbung in C++.
- (2) Es gibt Mehrfachvererbung in C++.
- (3) Es gibt Mehrfachvererbung in C++, aber nur, wenn alle Basisklassen mindestens eine virtuelle Funktion besitzen.
- (4) Es gibt Mehrfachvererbung in C++, aber nur, wenn alle Basisklassen mindestens eine rein-virtuelle Funktion besitzen.

e) Ihre Antwort: ____

- (1) Templates dienen dazu, abstrakte Klassen zu definieren.
- (2) Templates dienen dazu, auftretende Datentypen mit T abkürzen zu können.
- (3) Templates sind veraltet und werden in zukünftigen C++-Standards nicht mehr unterstützt.
- (4) Templates dienen dazu, gleichen Code für verschiedene Datentypen zu verwenden.

Aufgabe 5 (10 Punkte)

Schreiben Sie eine Funktion `sum`, die die Zahlen `1..n` aufsummiert und das Ergebnis geeignet zurückgibt. Der Rückgabebetyp dieser Funktion ist `void`, einer der Parameter soll die Zahl `n` sein.

- a) Geben Sie eine geeignete Signatur der Funktion an.

(2 Punkte)

- b) Geben Sie eine Implementierung an. Für den Fall, dass `n` negativ ist, werfen Sie eine geeignete Ausnahme.

(6 Punkte)

- c) Geben Sie ein Code-Beispiel an, wie der Benutzer diese Funktion aufruft.

(2 Punkte)

Aufgabe 6 (12 Punkte)

- a) Schreiben Sie eine Template-Funktion `swap` zur Vertauschung des Inhaltes zweier Variablen vom Template-Typ `T`.

(6 Punkte)

- b) Schreiben Sie eine Template-Klasse `Helper` (mit template-Parameter `C`) und definieren Sie dort eine statische Funktion `Potenz`, die die n 'te Potenz zu einer Basis `b` vom Datentyp `C` berechnet und das Ergebnis im gleichen Datentyp zurückgibt. Gehen Sie davon aus, dass `n` und die Basis `b` übergeben werden, dass für den Datentyp `C` die verwendeten arithmetischen Operationen definiert sind und dass $n > 0$ gilt.

(4 Punkte)

- c) Geben Sie an, wie Sie die Funktion `Potenz` für $n=5$ und einer Basis `b=2.0` vom Typ `double` aufrufen.

(2 Punkte)

Aufgabe 7 (12 Punkte)

Erklären Sie bitte folgende Begriffe und ggf. die wesentlichen Unterschiede *stichwortartig*. Verdeutlichen Sie Ihre Erklärung durch ein Beispiel zum Einsatz oder eine Erläuterung, wozu man den Begriff braucht.

a) Deklaration und Definition.

(4 Punkte)

b) Abstrakte Basisklasse.

(4 Punkte)

c) Pointer und Referenz.

(4 Punkte)

Aufgabe 8 (12 Punkte)

- a) Schreiben Sie eine Basisklasse `Figur`, die eine Position (x,y, ganzzahlig) sowie einen statischen Zähler enthält, der die Anzahl aktuell existierender Objekte enthält. Definieren Sie genau einen Konstruktor, der folgenden Code versteht:

```
Figur f1,f2(2),f3(3,4);
```

einen Copy-Konstruktor sowie den Destruktor. Wählen Sie geeignete Default-Werte. Ergänzen Sie Ihre Klasse `Figur` um eine virtuelle Funktion `Area`, die die belegte Fläche zurückgeben soll und natürlich den Ausgabeoperator `operator<<`.

(6 Punkte)

- b) Leiten Sie von `Figur` eine Klasse `Kreis` ab. Die Deklaration von Kreisobjekten sieht so aus:

```
Kreis k1,k2(1),k3(2,3),k4(2,3,3);
```

Der dritte Parameter ist der Radius, wählen Sie auch hier einen geeigneten Default-Wert. Implementieren Sie hier nur einen Konstruktor und die Funktion `Area`, so dass die Kreisfläche zurückgegeben wird ($A=3.14 \cdot \text{Radius}^2$).

(6 Punkte)

Aufgabe 9 (8 Punkte)

Sie wollen Ihre Klasse `C` bzw. Objekte `a, b, c` dieser Klasse entsprechend nachfolgender Code-Schnipsel verwenden. Geben Sie bitte (nur) die Signatur eines geeigneten Operators an, so dass das gelingt. Berücksichtigen Sie, dass alle Objekte der Klasse `C` sehr groß sind und nicht immer verändert werden.

a) Ausgabe: `std::cout << c << std::endl;`

(2 Punkte)

b) Minus-Gleich: `a -= b;`

(2 Punkte)

c) Minus (nur der!): `c = a - b;`

(2 Punkte)

d) Index-Zugriff: `b[2] = 12; // 12 ist ein Integer`

(2 Punkte)