

Aufgabe 1 (1+1+1+1+2+4 Punkte)

- a.) Wie viele Argumente sind in der Definition eines unären überladenen Operators erforderlich?
- b.) Nennen Sie zwei Operatoren in C++ die nicht überladen werden dürfen.
- c.) Erläutern Sie die Bedeutung des this-Zeigers.
- d.) Es sei A eine Klasse ohne parameterlosen Konstruktor. Was ist bei der Ableitung einer Klasse B von der Klasse A bzgl. der B-Konstruktoren zu beachten?
- e.) Was verstehen Sie unter einer abstrakten Klasse? Erläutern Sie, wie programmtechnisch eine Klasse zu einer abstrakten definiert wird. Geben Sie ein Beispiel!
- f.) Gegeben sei folgende Funktion zum Tauschen von double -Werten im Zusammenhang mit einem Sortieralgorithmus:

```
void tausch (double& a, double& b)
{ double temp = a;
  a = b;
  b = temp;
}
```

Realisieren Sie die Funktion als Template, so dass Typunabhängigkeit für die Funktionsargumente erreicht wird.

Aufgabe 2 (10 Punkte)

```
#include <iostream>
using namespace std;

class CPolygon {
protected:
    int width, height;
public:
    void set_values (int a, int b)
    { width=a; height=b; }
    virtual int area ()
    { return (0); }
};

class CRectangle: public CPolygon {
public:
    int area ()
    { return (width * height); }
};

class CTriangle: public CPolygon {
public:
    int area ()
    { return (width * height / 2); }
};

int main () {
    CRectangle rect;
    CTriangle trgl;
    CPolygon poly;
    CPolygon * ppoly1 = &rect;
    CPolygon * ppoly2 = &trgl;
    CPolygon * ppoly3 = &poly;
    ppoly1->set_values (4,5);
    ppoly2->set_values (4,5);
    ppoly3->set_values (4,5);
    cout << ppoly1->area() << endl;
    cout << ppoly2->area() << endl;
    cout << ppoly3->area() << endl;
    cin.get();
    return 0;
}
```

a.) Erklären Sie kurz was im Programm passiert.

b.) Welche Ausgaben hat das Programm

Aufgabe 3 (2+8 Punkte)

```

class X {
public:
    void A() {}
};

class Y {
public:
    void B() {}
};

class Z {
public:
    void C() {}
};

class Q : public X, protected Y, private Z {
public:
    void Test()
    {
        A(); // OK ? 
        B(); // OK ? 
        C(); // OK ? 
    }
};

class R : public Q {
public:
    void Test2()
    {
        A(); // OK ? 
        B(); // OK ? 
        C(); // OK ? 

        Q t;
        Y *y = &t // OK ? 
        Z *z = &t // OK ? 
    }
};

```

a.) Erklären Sie kurz was im Programm passiert.

b.) Welche von der „OK ?“ gekennzeichneten Stellen sind richtig, und wieso?

Aufgabe 4 (10 Punkte)

```
# include < iostream >

using namespace std ;

int main ( void )
{
    int zahl , teiler , limit ;

    cout << " Obergrenze eingeben : ";
    cin >> limit ;
    cout << "Ausgabe: " << limit << ":\n";

    for ( zahl =2; zahl < limit ; zahl ++ )
    {
        teiler = 2;
        while ( zahl % teiler != 0 ) teiler ++;
        if ( teiler == zahl ) cout << zahl << " ";
    }
    cout<<endl;
    system("pause");
    return 0;
}
```

- a.) Welche Ausgaben hat das Programm.
- b.) Erklären Sie was im Programm passiert.

Aufgabe 5 (10 Punkte – 1 Punkt je Antwort)

Bitte jeweils eine Antwort ankreuzen (je nach Fragestellung die richtige oder falsche):

a.) A sei die Oberklasse und B die (abgeleitete) Unterklasse. Welche Aussage stimmt:

- A) A besitzt meistens mehr Methoden und weniger Variablen als B
- B) A besitzt meistens weniger Methoden und dafür aber mehr Variablen als B
- C) Im UML-Diagramm zeigt die Spitze des Pfeils zwischen A und B auf A
- D) Im UML-Diagramm muss am Ende der Verbindungslinie zwischen A und B eine Raute sein

b.) In welchem Zusammenhang taucht das Konzept der virtuellen Klassen in der Vorlesung auf:

- A) Mehrfachvererbung
- B) Polymorphie
- C) Virtuelle Funktionen
- D) Online-Campus

c.) Wenn Sie beabsichtigen, eine Klasse A als Basisklasse für andere Klassen einzusetzen, dann gilt:

- A) die abgeleitete Klasse muss unbedingt einen eigenen Konstruktor definieren
- B) mindestens ein Element von A sollte protected sein
- C) der Destruktor von A sollte als virtual deklariert sein.
- D) die Klasse muss via class parent A deklariert werden

d.) Was ist richtig? Eine statische Klassenvariable ...

- A) ... heißt statisch, weil sie nicht verändert werden kann
- B) ... heißt statisch, weil sie auch dann existiert, wenn kein Objekt der Klasse angelegt ist
- C) ... kann immer direkt bei der Deklaration in der Klassendefinition vorinitialisiert werden
- D) ... kann nur als private deklariert werden

e.) Was ist hier richtig:

- A) Jede Mehrfachvererbung kann mit zwei IS-A-Beziehungen simuliert werden.
- B) Pointer und Referenzen auf Objekte verhalten sich im Kontext der Polymorphie gleich.
- C) zwischen Pointer und Referenzen gibt es überhaupt keine Unterschiede im Verhalten
- D) eine Referenz kann man mit ++ erhöhen, einen Pointer nicht

f.) Polymorphie tritt nicht auf bei Verwendung von

- A) Objektvariablen,
- B) Objektreferenzen,
- C) Objektzeigern,
- D) Zeigern auf Objekte von abgeleiteten Klassen.

g.) Die Schnittstelle („Interface“) einer Klasse besteht aus

- A) polymorphen Ringen,
- B) den protected-Elementen einer Klasse oder
- C) den public-Elementen einer Klasse oder
- D) zwei konzentrischen Kreisen.

h.) Welches der folgenden Statements läßt der Compiler nach der Deklaration von `const int p[3] = {1,2,3};` sicher noch zu:

- A) `*p = 3;`
- B) `p++;`
- C) `(*p)++;`
- D) `int q = *p.`

i.) Welche Antworten zur Operatorüberladung sind richtig?

- A) Operatoren können zusammen mit Objekten funktionieren
- B) Operatoren wird mehr aufgeladen, als diese handhaben können.
- C) Operatoren wird neue Bedeutung verliehen