

## Klausur: Programmieren 2

Hochschule für Angewandte Wissenschaften Hamburg  
 Fakultät Technik und Informatik, Department Informations- und Elektrotechnik  
 Prof. Dr. Robert Heß, 5.7.2013, Dauer: 90 Min.  
 Hilfsmittel: Vorlesungsunterlagen und C/C++ Einführungsbecher (z.B. Kernighan/Ritchie).  
 Jegliche Art von elektronischen Hilfsmitteln sind untersagt.

Ergebnis: ..... von 100 Punkten                      Note: ..... Punkte.

### Aufgabe 1 (10 Punkte)

Überlegen Sie für den folgenden Quellcode, welche der in der Tabelle gezeigten Zeilen syntaktisch korrekt sind, und wenn ja, welche Werte der Variable *a* zugewiesen werden. Die Zeilen in der Tabelle werden einzeln in den Quellcode kopiert und getestet.

```
int main()
{
    double x[]={ 0.1, 1.23, 2, 10 };
    double *y=x;
    double **z=&y;
    double a;

    /* Zeile aus der Tabelle */

    return 0;
}
```

Ausdruck	syntaktisch korrekt	Ergebnis
a = x[3];		
a = *y		
a = y[2];		
a = *++y;		
a = *y++;		
a = **z;		
a = *++*z;		
a = *(*z)++;		
a = *++z;		

**Aufgabe 2 (8 Punkte)**

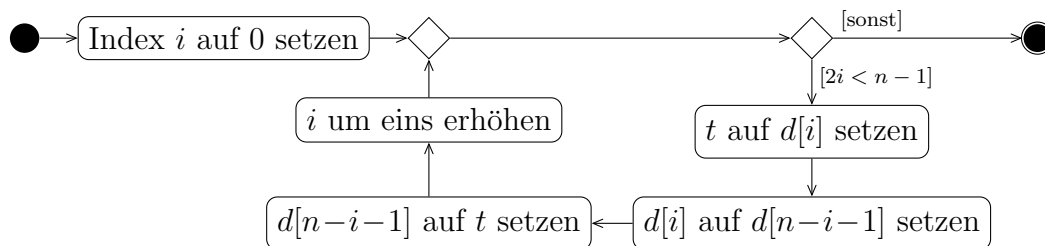
1. Mit welchen Funktionen wird in C Speicher reserviert?
2. Worauf muss beim Reservieren von Speicher geachtet werden?
3. Mit welcher Funktion wird in C Speicher freigegeben?
4. Erläutern Sie den Begriff *Memory leakage*.

**Aufgabe 3 (20 Punkte)**

Gegeben sei folgende Funktionsdeklaration:

```
void function(int *d, int n);
```

Die Arbeitsweise der Funktion ist durch folgendes Aktivitätsdiagramm beschrieben:



Die Funktion wird beispielhaft wie folgt aufgerufen:

```
int main()
{
    int a[]={1, 2, 3, 4, 5, 6};
    int i;

    for(i=0; i<6; i++) printf("%d%c", a[i], i<5?' ':'\n');
    function(a, 6);
    for(i=0; i<6; i++) printf("%d%c", a[i], i<5?' ':'\n');

    return 0;
}
```

1. Implementieren Sie auf einem separaten Blatt die Funktion.
2. Was wird bei dem beispielhaften Aufruf auf dem Bildschirm ausgegeben?
3. Beschreiben Sie in einem kurzen Satz, was die Aufgabe der Funktion ist.

#### Aufgabe 4 (20 Punkte)

Erstellen Sie auf einem separaten Blatt ein Programm `main()`, welches Zahlen  $1, \frac{1}{2}, \frac{1}{4}, \dots, \frac{1}{64}$  vom Typ `double` binär in eine Datei mit Namen *Zahlen.dat* schreibt.

#### Aufgabe 5 (15 Punkte)

Betrachten Sie folgende Funktion:

```
int function(unsigned n)
{
    if(n>0) return function(n/10)+n%10;
    return n%10;
}
```

1. Welchen Wert gibt die Funktion bei einem Aufruf mit `function(12345u)`; zurück?
2. Geben Sie der Funktion einen sinnvollen Namen:

#### Aufgabe 6 (10 Punkte)

Erstellen Sie mit `typedef` und `struct` einen Datentyp für eine einfach verkettete Liste zum Speichern von dreidimensionalen mathematischen Vektoren  $(x, y, z)^T$ .

**Aufgabe 7 (10 Punkte)**

1. Wann wird der Konstruktor eines C++-Objekts aufgerufen?

---

2. Wann wird der Destruktor eines C++-Objekts aufgerufen?

---

3. In einem C++-Objekt befinden sich die Attribute und Methoden in den Bereichen *public*, *protected* und *private*. Von wo können auf diese Elemente zugegriffen werden?

Zugriffsbereich	innerhalb des Objekts		außerhalb des Objekts	
public	möglich <input type="checkbox"/>	unmöglich <input type="checkbox"/>	möglich <input type="checkbox"/>	unmöglich <input type="checkbox"/>
protected	möglich <input type="checkbox"/>	unmöglich <input type="checkbox"/>	möglich <input type="checkbox"/>	unmöglich <input type="checkbox"/>
private	möglich <input type="checkbox"/>	unmöglich <input type="checkbox"/>	möglich <input type="checkbox"/>	unmöglich <input type="checkbox"/>

**Aufgabe 8 (7 Punkte)**

Fügen Sie folgender Klasse die Deklaration eines Operators zum Addieren komplexer Zahlen hinzu.

```
class cComplex {
public:
    double re; // Realteil
    double im; // Imaginärteil
    cComplex(); // Konstruktor
    // Ihr Operator zum Addieren von komplexen Zahlen:
```

};