

## Vorlesung “Objektorientierte Programmierung” — Probeklausur II —

Bitte geben Sie diese Probeklausur anonym ab. Damit Sie die Klausur bei der Rückgabe leichter finden können, tragen Sie bitte in die dafür vorgesehenen Kästchen eine dreistellige, zufällige Zahl ein. Merken Sie sich diese Zahl.

Zahl zur Identifikation bei Rückgabe:

Aufgabe	Punkte	von (Max.)	Zeit
1 (Rekursive Funktion)		1	5 min
2 (Lokale Variablen, Parameterübergabe)		1	5 min
3 (Strings, Pointer)		1	5 min
4 (Programm)		5	15 min
5 (Fehlersuche, Strukturen)		2	5 min
Summe		10	35 min

### Hinweise:

- Bearbeitungsdauer: 40 Minuten
- Skript, Bücher, Notizen sind erlaubt. Notebooks, PDAs, etc. dürfen nicht verwendet werden. Natürlich dürfen Sie auch nicht mit dem Nachbarn sprechen oder Papiere und andere Dinge austauschen. Notfalls rufen Sie eine Aufsichtsperson zur Kontrolle.
- Bitte benutzen Sie den vorgegebenen Platz. Wenn Sie auf die Rückseite ausweichen müssen, markieren Sie klar, daß es eine Fortsetzung gibt.
- Fragen Sie, wenn Ihnen eine Aufgabe nicht klar ist!
- **ACHTUNG!** Diese Probeklausur ist möglicherweise einfacher als die eigentliche Klausur. Falls Sie mit der Probeklausur keine Schwierigkeiten haben, heißt das nicht, daß Sie sich für die eigentliche Klausur nicht vorbereiten müssen.

### Umfrage:

- Wieviel Prozent des Vorlesungsstoffs sind für Sie neu?  
 100%     75%     50%     25%     0%
- Besuchen Sie regelmäßig die Vorlesung?  
 fast immer     öfters     fast nie

**Aufgabe 1 (Rekursive Funktion)****1 Punkt**

Was gibt das folgende Programm aus?

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

void f(int n)
{
    if(n > 0) {
        cout << -n << ' ';
        f(n - 1);
        cout << n << ' ';
    }
    else
        cout << "* ";
}

int main()
{
    f(2);

    return 0;
}
```

Ausgabe: \_\_\_\_\_

**Aufgabe 2 (Lokale Variablen, Parameterübergabe)****1 Punkt**

Was gibt das folgende Programm aus?

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

int n = 1;

int f(int n)
{
    n++;
    n = n * 2;
    if(n > 20)
        n = n << 3;
    n *= 0;
    return n;
}

void g(int *p)
{
    (*p)--;
    n += 5;
    cout << n << ' ';
}

int main()
{
    int n = 4;
    int k = f(n);
    g(&n);
    cout << n;
    return k;
}
```

Ausgabe: \_\_\_\_\_

**Aufgabe 3 (Strings, Pointer)****1 Punkt**

Was gibt dieses Programm aus?

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

int main()
{
    const char *p = "123";
    char a[10];
    char *q = a;
    while(*p) {
        *q = *p;
        p++;
        q++;
    }
    a[1] = '\\0';
    cout << a;
    return 0;
}
```

Ausgabe: \_\_\_\_\_

**Aufgabe 4 (Programm)****5 Punkte**

Schreiben Sie ein Programm, das 10 ganze Zahlen (`int`-Werte) einliest, und dann “ja” ausgibt, falls eine Zahl in der Folge doppelt vorgekommen ist, und “nein” sonst. Z.B. könnte die Eingabe sein:

3 45 7 -6 98 4 55 0 1 2

In diesem Fall soll “nein” ausgegeben werden, da keine Zahl doppelt vorkommt. Bei der Eingabe

0 1 2 3 4 2 5 6 7 8

soll dagegen “ja” ausgegeben werden, weil die 2 doppelt ist. Die Ausgabe soll erst erfolgen, nachdem alle Zahlen eingelesen wurden. Fehler bei der Eingabe brauchen Sie nicht zu behandeln. Es soll leicht möglich sein, die Zahl 10 durch eine andere Zahl zu ersetzen. Bemühen Sie sich um Verständlichkeit und guten Programmierstil (es gibt ggf. Punktabzüge). Schreiben Sie das Programm vollständig und ohne Syntaxfehler.

**Aufgabe 5 (Fehlersuche, Strukturen)****2 Punkte**

Das folgende Programm enthält (mindestens) 3 Fehler, die der Compiler finden sollte. Bitte geben Sie zwei dieser Fehler an (es ist möglicherweise schwierig, alle drei zu entdecken). Sie bekommen keinen Extrapunkt, wenn Sie alle drei finden. Falls Sie mehr als zwei Fehler angeben, werden nur die ersten beiden gewertet. Geben Sie bitte die Zeilennummer mit an, in der der Compiler den Fehler finden müßte, wenn er das Programm von oben nach unten liest.

```
(1) #include <iostream>
(2) using namespace std;
(3)
(4) struct s1 { int i; };
(5) struct s2 { int i; };
(6)
(7) int main()
(8) {
(9)     int i;
(10)    s1 x1;
(11)    s2 x2;
(12)    cin >> i;
(13)    x1.i = i;
(14)    x2 = x1;
(15)    int *p = &x2;
(16)    if(x1->i == 0)
(17)        cout << 1;
(18)    else
(19)        cout << 0;
(20)    return 0;
(21) }
```

Fehler 1:

Zeile: \_\_\_\_\_

Begründung: \_\_\_\_\_

Fehler 2:

Zeile: \_\_\_\_\_

Begründung: \_\_\_\_\_