

**Aufgabe 1: Grundlagen****(5 Punkte)**

Welche der folgenden Behauptungen über die Programmiersprache Go sind wahr, welche falsch?

Behauptung	wahr	falsch
<code>int16</code> ist ein Datentyp in Go.		
<code>float64</code> ist ein Datentyp in Go.		
Eine Funktionssignatur sagt nichts über den Rückgabotyp der Funktion aus.		
Bei der Definition von Variablen muss der Datentyp immer feststehen.		
<code>range</code> -Schleifen haben keinen Schleifenzähler.		

*Anmerkung:* Korrekt angekreuzte Zeilen geben einen Punkt, für falsch angekreuzte Zeilen wird ein Punkt abgezogen.

**Aufgabe 2: Signaturen****(10 Punkte)**

Betrachten Sie das folgende Programmfragment:

```
1  x1 := Foo1("Hallo", 15)
2  x2 := Foo2(x1)
3  x3 := Foo3(127, x2)
4  if x2 {
5      x3 = append(x3, Foo1("Welt", x1))
6  }
7  x1 += Foo4(x2, true)
8  Foo5(x2 && x1 != Foo4(x2, x2))
9  return x2 && !(x1 > x3[0])
```

Welche Signaturen haben die Funktionen `Foo1` bis `Foo5`? Welchen Rückgabetyt liefert das `return`?

*Anmerkung:* Die *Signatur* einer Funktion ist die erste Zeile, in der die Argument- und Rückgabetypen definiert werden. Hier ist also gefragt, welche Typen die Funktionen erwarten und liefern. Sie können davon ausgehen, dass Funktionen, deren Ergebnis nicht verwendet wird, auch keinen Rückgabetyt haben.

**Aufgabe 3: Fehlersuche: Compilerfehler****(10 Punkte)**

Der folgende Code enthält eine Reihe an Fehlern, durch die er nicht compiliert. Markieren Sie alle Zeilen, die einen Fehler enthalten und erläutern Sie kurz, was jeweils falsch ist.

```
1 package fehlersuche1
2
3 import "fmt"
4
5 func Foo(x int) int {
6     return x := 3
7 }
8
9 func Bar(x, y int) string {
10    return fmt.Sprintf(Foo(5))
11 }
12
13 Func FooBar() {
14     s := 'Huhu'
15     for x := range ([]int{1,2,3,4,5}) {
16         fmt.Println(x)
17     }
18     s += y
19 }
```

**Hinweis:** Es geht hier nur um Syntaxfehler. Für jede falsch markierte Zeile gibt es Punktabzug!

**Aufgabe 4: Fehlersuche: Inhaltliche Fehler****(5 Punkte)**

Die folgende Funktion ist zwar syntaktisch korrekt, sie erfüllt aber nicht ihre Aufgabe. Erläutern Sie den/die Fehler und machen Sie einen Vorschlag zur Korrektur.

```
1 // Sorted liefert true, falls die Liste aufsteigend sortiert ist.
2 func Sorted(list []int) bool {
3     for _, el := range list[1:] {
4         if el < list[el] {
5             return false
6         }
7     }
8     return true
9 }
```

*Anmerkung:* Ihre Korrektur muss nicht syntaktisch korrekt sein. Eine Erklärung in Worten genügt.

**Aufgabe 5: Programmverständnis****(5 Punkte)**

Erläutern Sie, was die Funktion `Foo` im folgenden Programmfragment berechnet. Geben Sie eine möglichst allgemeine bzw. abstrakte Erklärung an.

```
1 func Bar(n, i int) int {
2     if n < 0 || i == 0 {
3         return -1
4     }
5     if i*i == n {
6         return i
7     }
8     return Bar(n, i-1)
9 }
10
11 func Foo(n int) int {
12     return Bar(n, n)
13 }
```

**Aufgabe 6: Rekursion****(10 Punkte)**

Betrachten Sie die folgende Funktion:

```
1 func Foo(n, c int) int {  
2     if n == 0 {  
3         return 0  
4     }  
5     return Foo(n/10, c+1) + c*(n%10)  
6 }
```

Beschreiben Sie in Worten, was die Funktion berechnet.

Berechnen Sie außerdem beispielhaft die Werte der Funktion für  $n = 10$ ,  $n = 11$  und  $n = 201$  mit  $c = 1$ . Geben Sie dabei die Zwischenergebnisse der rekursiven Aufrufe mit an.