

Bitte schicken Sie Ihre Abgabe des Übungsblattes an die folgende, bereits bekannte E-Mail-Adresse **prog1@cl.uni-heidelberg.de** bis zum **27.05.2018 , 23:59 Uhr** und halten Sie folgende Formalia für die Abgabe ein, um keinen Punktabzug zu erhalten:

- Betreff: [Prog1] Abgabe Übungsblatt [zweistellige Nummer], [Vorname] [Nachname]
- Dateiname: [nachname]_[vorname]_[Übungsblattnummer].[dateiendung]
- Im Betreff sind die Anfangsbuchstaben jeweils groß zu schreiben und in den Dateinamen nur Kleinbuchstaben zu verwenden.
- Falls Sie mehrere Dateien mit gleicher Dateiendung verwenden sollten, dann benutzen Sie Kleinbuchstaben zur weiteren Unterscheidung,
z. B. [nachname]_[vorname]_[Übungsblattnummer][buchstabe].[dateiendung]
- Trennen Sie Doppelnamen bitte stets mit einem Bindestrich (auch im Dateinamen), z. B. [Prog1] Abgabe Übungsblatt 01, Max Mustermann-Müller
- Zusätzliche Namensteile lassen Sie bitte weg oder trennen diese ebenfalls mit einem Bindestrich, z. B. von-Mustermann-Müller
- Bitte geben Sie nur Python-Dateien (.py) für Ihren Code ab, für digitale schriftliche Abgaben bitte nur PDF-Dateien oder einfache Text-Dateien (.txt).

E-Mail (Muster):

Von: mustermann@cl.uni-heidelberg.de
An: prog1@cl.uni-heidelberg.de
Betreff: [Prog1] Abgabe Übungsblatt 01, Max Mustermann

Liebe Tutoren,

anbei ist meine Abgabe für das erste Übungsblatt.

Schöne Grüße
Max Friedrich Karl-Heinz von Mustermann

Anlage 1: mustermann_max_01.pdf
Anlage 2: mustermann_max_01a.py
Anlage 3: mustermann_max_01b.py

Laden Sie sich das Programm/Modul http://www.cl.uni-heidelberg.de/courses/ss18/prog1/material/worst_code.py herunter und speichern Sie es in ein Verzeichnis ab, von dem Sie dieses in anderen Programmen importieren können. Das Modul soll die Möglichkeit geben, zu überprüfen, ob eine Liste mit Integern bereits aufsteigend sortiert ist.

```
1 In [1]: import worst_code
2 ListIsSorted started
3 In [2]: example_list_1 = [1, 5, 17, 22, 100]
4 In [3]: example_list_2 = [3, 1, 44, 10, 655]
5 In [4]: print(worst_code.ListIsSorted(example_list_1))
6 True
7 In [5]: print(worst_code.ListIsSorted(example_list_2))
8 False
```

Programmcode 1: Beispielimport

1. (2 Punkte) Der Code scheint also zu funktionieren und das zu tun, was er soll. Ist das aber wirklich so? Finden Sie zwei (logisch verschiedene) Gegenbeispiele (Liste von Zahlen), die deutlich aufzeigen, warum das nicht so ist und erläutern Sie das kurz.
2. (2 Punkte) Der Code läuft nicht nur nicht immer wie erwartet, sondern sieht auch nicht so schön aus. Finden Sie mindestens vier Stellen, bei denen *The Zen of Python* mit dem Code vermutlich nicht zufrieden wäre.
3. (1 Punkt) Verbessern Sie lediglich die Dokumentation des Moduls (ohne die Code-Bestandteile zu verbessern). Speichern und geben Sie dieses Modul dann auch so wieder ab (also bitte auch als *worst_code.py* abgeben).
4. (3 Punkte) Schreiben Sie ein neues Modul (Auch hier ist der Dateiname nicht wie in den Formalia angeben, sondern einfach *good_code.py*), welches genau eine Funktion mit der Signatur "is_sorted(list array) -> bool" hat. Können Sie die Überprüfung, ob eine Liste von Zahlen aufsteigend sortiert ist, besser hinbekommen als im Modul *worst_code*?
5. (2 Punkte) Finden Sie zwei verschiedene Lösungen für dieses Problem im Internet. Sie müssen den Code nicht unbedingt verstehen. Die Signaturen sollten Sie aber folgendermaßen anpassen: "is_sorted_1(list array) -> bool" und "is_sorted_2(list array) -> bool". Die Lösungen sollen für Python ≥ 3.5 funktionieren. Schreiben Sie diese Funktionen einfach noch in *good_code.py* hinzu. Dokumentieren gehört natürlich dazu!

Aufgabe:	1	2	3	4	5	Summe:
Punkte:	2	2	1	3	2	10
Ergebnis:						