

LU02.L11 - Immutable Dataclass

student.py

```
"""
The Student class is a dataclass that represents a student
with a name, grades, and a graduation status.
"""

from dataclasses import dataclass, field
from typing import List

@dataclass(frozen=True)
class Student:
    """
    A class representing a student with a name, grades, and a
    graduation status.
    """

    name: str
    grades: List[int] = field(default_factory=list)
    graduated: bool = False
```

main.py

```
"""
This is the main module of the application.
"""

from student import Student

def add_grade(student, grade):
    """
    Returns a new Student instance with the added grade.
    """
    new_grades = student.grades + [grade]
    return Student(name=student.name, grades=new_grades,
graduated=student.graduated)

def calculate_average(student):
    """
    Returns the average of the student's grades.
    """
    if not student.grades:
        return 0.0
```

```
    return round(sum(student.grades) / len(student.grades), 2)

def graduate_student(student):
    """
    Graduates the student if the average grade is 70 or above.
    """
    average = calculate_average(student)
    if average >= 70:
        return Student(name=student.name, grades=student.grades,
graduated=True)
    return student

if __name__ == '__main__':
    stu = Student(name='John Doe')
    stu = add_grade(stu, 85)
    stu = add_grade(stu, 75)
    stu = add_grade(stu, 60)

    print(f'Noten: {stu.grades}')
    avg = calculate_average(stu)
    print(f'Durchschnitt: {avg}')

    stu = graduate_student(stu)
    print(f'Absolviert: {stu.graduated}')
```



© Kevin Maurizi

From:
<https://wiki.bzz.ch/> - **BZZ - Modulwiki**

Permanent link:
<https://wiki.bzz.ch/modul/m323/learningunits/lu02/loesungen/dataclass2>

Last update: **2024/08/28 11:46**

