

Lecture 10: Classes, part 2

Morten Rieger Hannemose, Vedrana Andersen Dahl

Fall 2023

$$f(x+\Delta x) = \sum_{i=0}^{\infty} \frac{(\Delta x)^i}{i!} f^{(i)}(x)$$

Today's lecture

1. Continuation about OOP (5 min)
2. A coding demo (40 min)

$$f(x+\Delta x) = \sum_{i=0}^{\infty} \frac{(\Delta x)^i}{i!} f^{(i)}(x)$$

Continuing where we left off

MyTime

```
1 class MyTime:
2
3     def __init__(self, hours, minutes):
4         self.hours = hours
5         self.minutes = minutes
6
7     def print_time(self):
8         print(f'{self.hours:02}:{self.minutes:02}')
9
10    def increment_hours(self):
11        self.hours += 1
12        if self.hours == 24:
13            self.hours = 0
14
15    def increment_minutes(self):
16        self.minutes += 1
17        if self.minutes == 60:
18            self.minutes = 0
19            self.increment_hours()
20
21 my_time = MyTime(23, 55)
22 for i in range(10):
23     my_time.increment_minutes()
24     my_time.print_time()
```

Recall from last time

- ▶ Attributes and methods bundled together. This is referred to as *encapsulation* and sometimes the access to attributes is restricted.
- ▶ Class definition is concerned with the template for objects. Actual objects are made through instantiation.
- ▶ Methods have special syntax, in particular the `self` attribute.

Today we continue working with OOP

- ▶ The `__str__` method
- ▶ Inheritance: defining a child class for `MyTime`
- ▶ Operator overloading shown on the `__add__` method

Code shown live during lecture

MyTimeSeconds

```
1 class MyTimeSeconds(MyTime):
2     def __init__(self, hours, minutes, seconds):
3         super().__init__(hours, minutes)
4         self.seconds = seconds
5     def __str__(self):
6         return super().__str__() + f":{self.seconds
7         :02}"
8     def __add__(self, other):
9         hours = self.hours + other.hours
10        minutes = self.minutes + other.minutes
11        seconds = self.seconds + other.seconds
12        if seconds >= 60:
13            seconds -= 60
14            minutes += 1
15        if minutes >= 60:
16            minutes -= 60
17            hours += 1
18        if hours >= 24:
19            hours -= 24
20        return MyTimeSeconds(hours, minutes, seconds
21
22 my_time1 = MyTimeSeconds(2, 30, 59)
23 my_time2 = MyTimeSeconds(2, 30, 59)
24 print("The time is", my_time1 + my_time2)
```

MyTimeUS

```
1 class MyTimeUS(MyTimeSeconds):
2     def __str__(self):
3         if self.hours >= 12:
4             AMPM = "PM"
5         else:
6             AMPM = "AM"
7         new_hours = (self.hours - 1) % 12 + 1
8         return f"{new_hours:02}:{self.minutes:02}" +
9             " " + AMPM
10 my_time = MyTimeUS(11,00, 00)
11 print(my_time)
12 my_time.increment_hours()
13 print(my_time)
14 my_time.increment_hours()
15 print(my_time)
```