CPSC 103

Introduction to Systematic Program Design

2021S

Lecture: Module 4 - Compound

Ashish Chopra

20 May, 2021

Announcements

1. Syllabus Quiz due Tomorrow.
2. Tutorial Resubmission is open.
3. Project
    1. If you are re-taking this course, you cannot use the same project you submitted before.
       It is considered as Academic Misconduct.
4. Midterm
    1. Date: Fri May 28, 7pm PDT
    2. Extra office hours next week to support midterm preparation (will be announced on Piazza)
    3. Midterm Review session by TAs (to be announced on Piazza)
    4. I will be posting a detailed post about FAQ related to Midterm by this weekend.
    5. “Sample Exams” module will be available on Canvas by tomorrow
5. Module 5 and Module 6 open for pre-class readings.

Recap

1. Design Data Definitions for Non-Primitive Data Types (using HtDD)
    1. Simple Atomic (e.g., CountryName)
    2. Interval (e.g., Temperature)
    3. Optional (e.g., CountdownTimer, SensorReading)
    4. Enumeration (e.g., Grade, TrafficLight)
2. How to Design functions which operate on non-primitive data types
    1. Robotic Wheelchair Problem
    2. Next Traffic Light Problem

Recap

How to Design Data
(HtDD) Recipe

HtDD recipe is about designing data definitions of user-defined types.

HtDD recipe consists of the following steps:

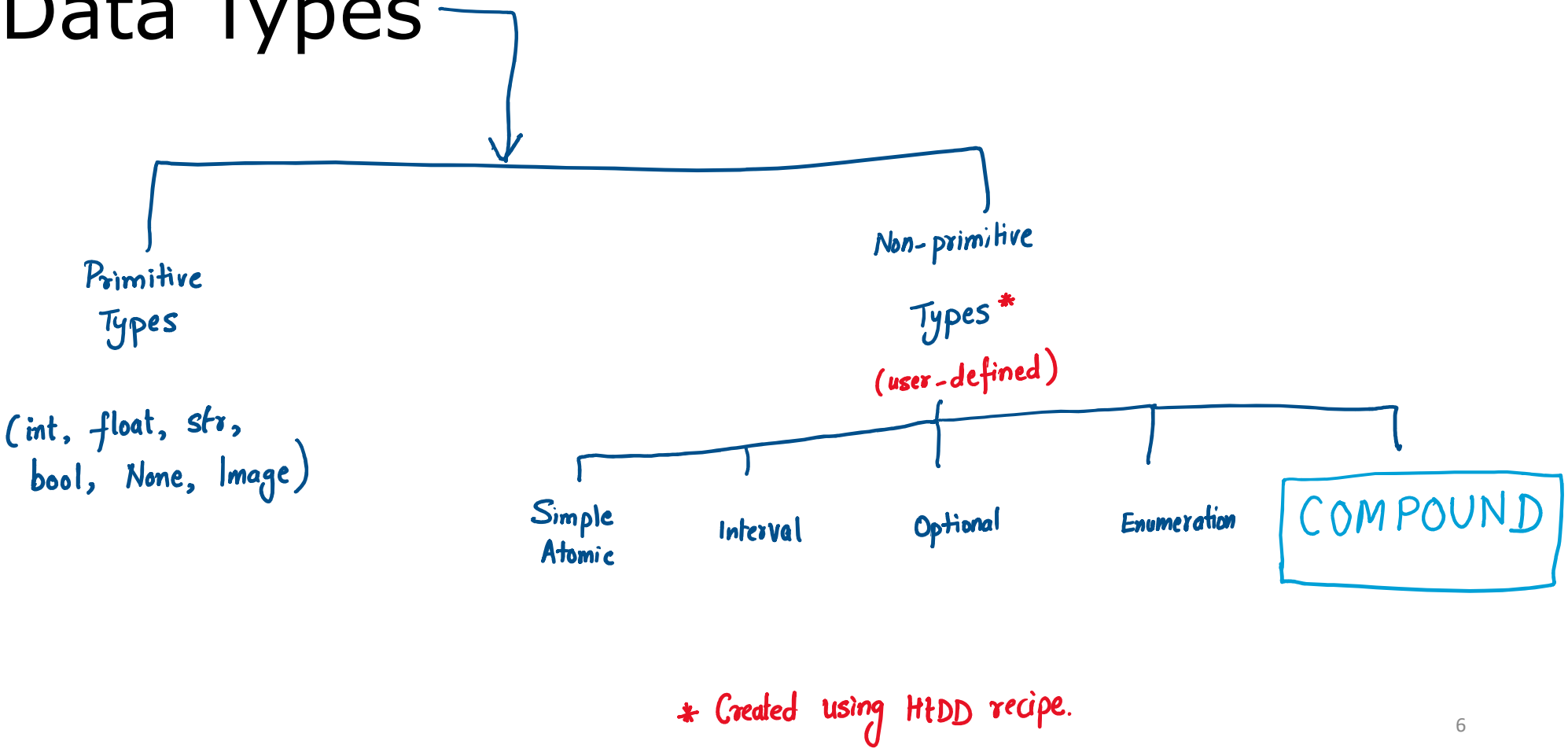
1. A **data type definition** with type comments where Python's types are not specific enough.
2. An **interpretation comment** that describes the correspondence between information and data.
3. One or more **examples** of the data.
4. A **template** for a one-argument function operating on data of this type.

```
Temperature = float
# interp. the air temperature in degrees Celsius

T1 = 0.0
T2 = -24.5
T3 = 10.2

@typecheck
# template based on Atomic
def fn_for_temperature(t: Temperature) -> ...:
    return ...(t)
```

Learning Goals

1. Compound Data
    1. Identify Problem Domain Information that should be represented as compound data.
2. Design Data definitions for Compound
    1. Understand and write NamedTuple definitions.
    2. Use HtDD and Data Driven Templates with compound data.
3. Design Functions that take in and/or return compound data.

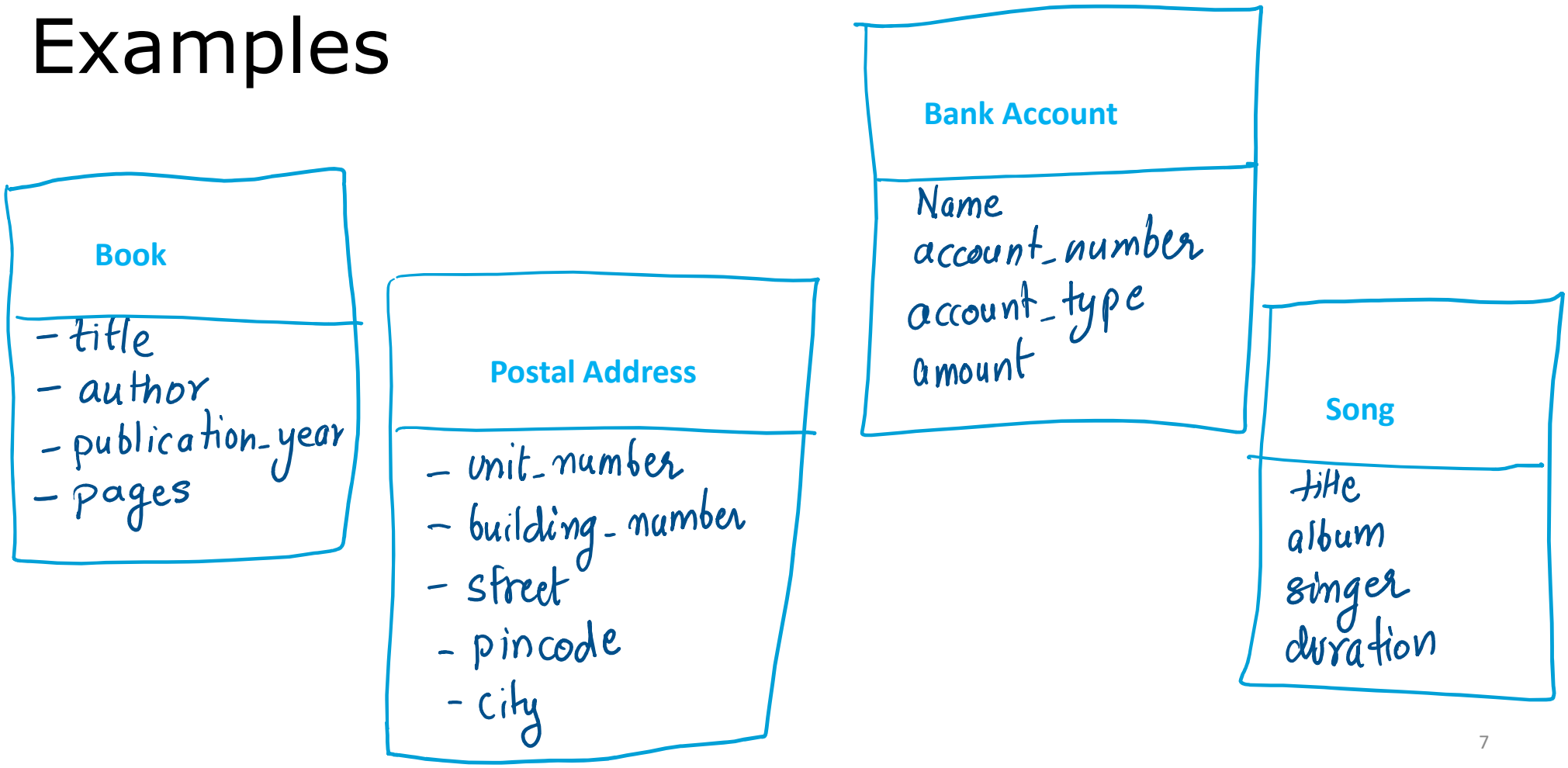
Data Types



* Created using HtDD recipe.

Compound

Examples



Assert Statements

For information which has a range, we specify it via comments., but how can we make sure that someone will not provide value outside the range?

(1) In the fn that takes a Song, we will check:
assert duration > 0

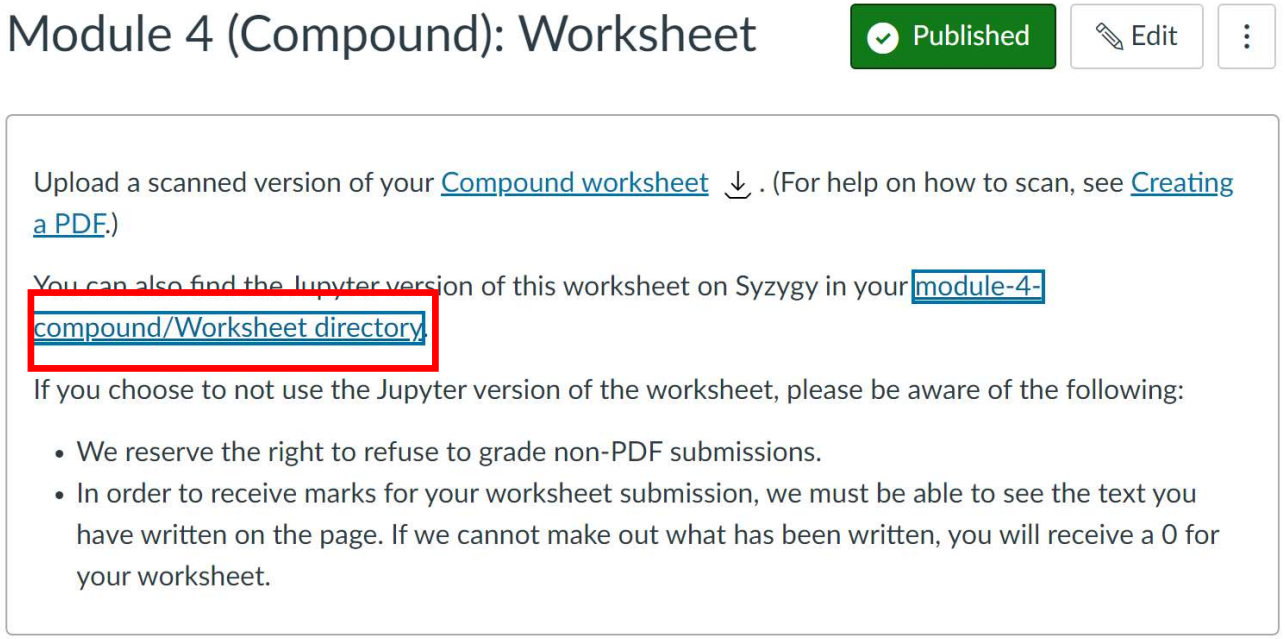
(2) In can_we_play() function, we will check if min_players and max_players are in range using assert.

Rest on Jupyter Notebook at:

Module4-compound > Lecture > Ashish > Lecture Python Notebook – Module 4 (Compound) – Blank.ipynb

Worksheet Activity Time!

Let's do
Question 1, 3 - 5



Worksheet Activity Time!

Let's do
Question 2, 6, 7, 8