CPSC 103

Introduction to Systematic Program Design

2021S

Lecture: Module 6 – One Task Per Function

Ashish Chopra

1 June, 2021

Announcements

1. Project
   1. Project TA Mentors are announced on Piazza.
   2. Start working on Project Milestone Submission.
2. Midterm marking on the way!
3. Module 8 is open for pre-class readings.

Recap

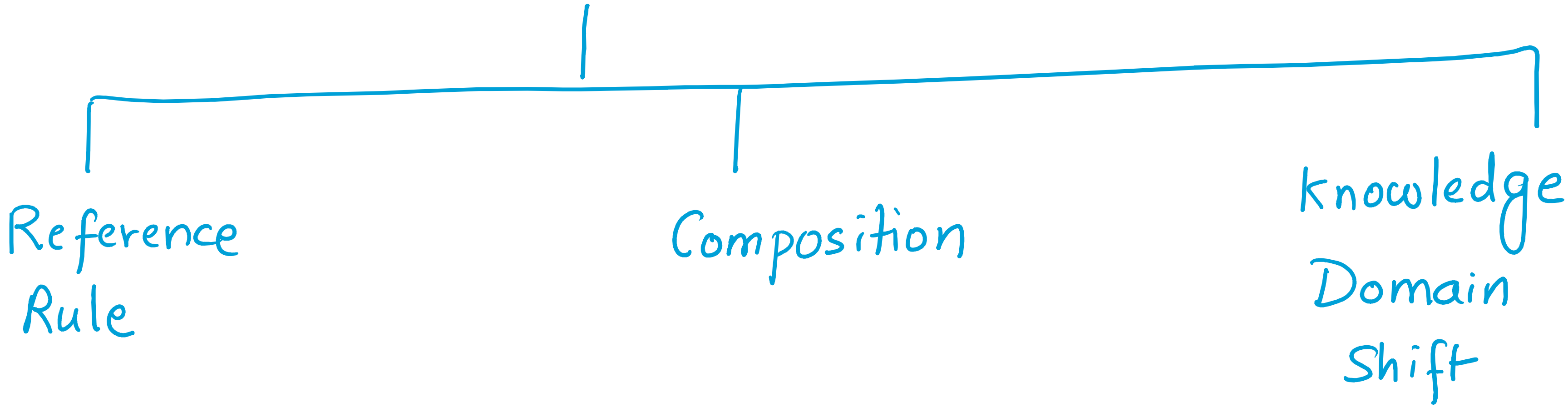
1. Arbitrary-Sized List of Compounds
    1. List[Recipe] ✓
    2. List[Book] ✓
    3. List[Song] ✓
2. Reference Rule *
    1. List[Book] references Book type (other non-primitive data type)
3. Helper Functions

Learning Goals

1. All about Helper Functions (a.k.a. One Task Per Function)
    1. Design functions that each focus on one task only.
    2. We discuss few helper rules that guides us in identifying the need of helper functions.
    3. Practice, Practice, Practice!

Helper Rules

Each function should have one simple task that it does, and it should hand off to other functions any additional tasks

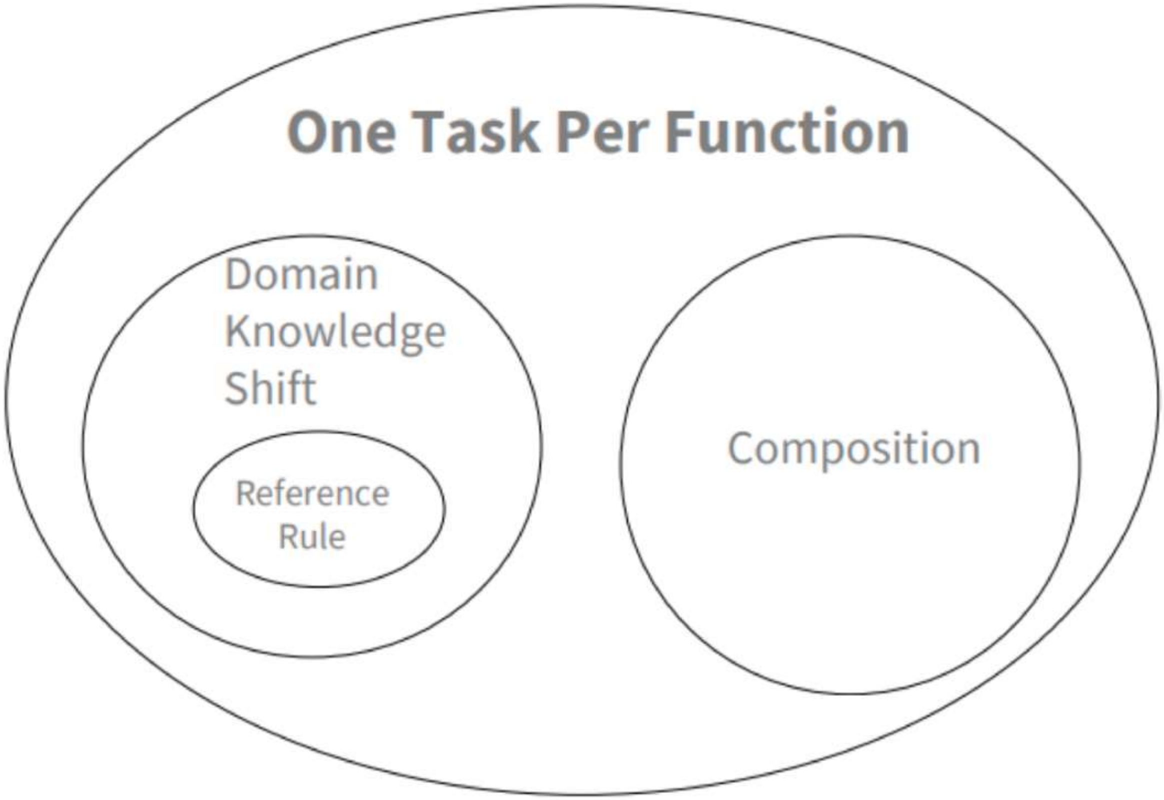


Reference Rule

Composition

Knowledge Domain Shift

Helper Function Rules



1. Reference Rule

Use a helper function at references to other non-primitive data definitions (this will be in the template).

When we are solving a problem using a non-primitive data A and this data uses another non-primitive data B. We usually put this on the template (remember previous class?).

Example: When solving something on a list (non-primitive data) of Book (non-primitive data).

2. Composition

Use a helper function for each distinct and complete operation that must be performed on the input data.

- When we are solving a problem that requires several steps that are sequential.
- We usually want our functions to be small and have a clear purpose.
- When in doubt, always create a new function! It is better to have more functions than one complex function.

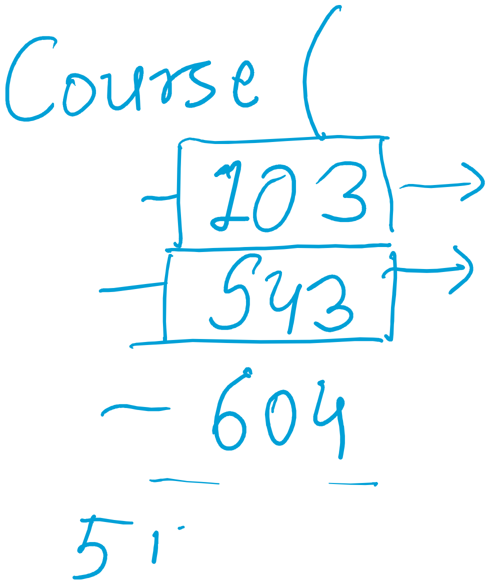
Example: Calculating the average of a list of integers. We can first calculate the sum, then count the number of elements, then finally calculate the average.

3. Knowledge Domain Shift

Use a helper function if a subtask involves knowledge about a type that is not taken as input by this function.

- It happens usually when we have a primitive data type representing a more complex data.

- This is a rare case and you almost won’t see it in this course.



Example: Example: When using integers to represent months, we are not using the value as **regular numbers** (adding, multiplying), so we may want a different function to deal with this type of data.

Calling Helper Functions

There are two ways of calling functions that sequentially use each others returns values. There are no better or worse way, either are the same.

① DIRECTLY CALLING

```
return second_function(first_function(parameter))
```

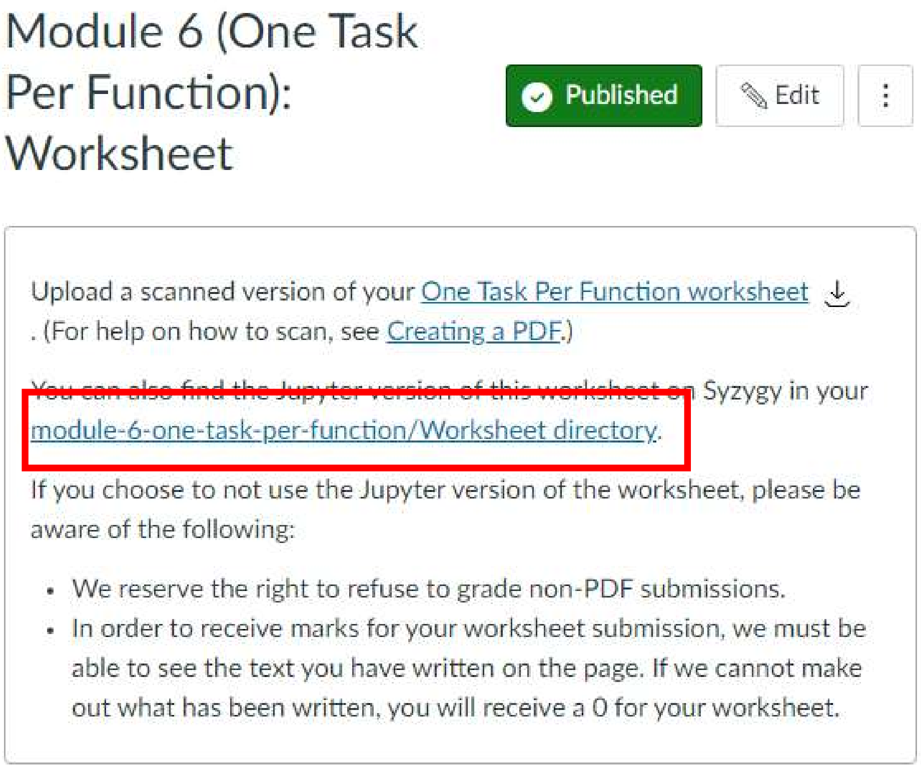
② USING VARIABLES

```
first_step = first_function(parameter)
second_step = second_function(first_step)
return second_step
```

Worksheet Activity Time!

Let’s do Question 1 - 2

Let’s meet back in 15 minutes



Module 6 (One Task Per Function): Worksheet

Upload a scanned version of your One Task Per Function worksheet . (For help on how to scan, see Creating a PDF.)

You can also find the Jupyter version of this worksheet on Syzygy in your module-6-one-task-per-function/Worksheet directory.

If you choose to not use the Jupyter version of the worksheet, please be aware of the following:

- We reserve the right to refuse to grade non-PDF submissions.
- In order to receive marks for your worksheet submission, we must be able to see the text you have written on the page. If we cannot make out what has been written, you will receive a 0 for your worksheet.