ENGINEERING
TEXAS A&M UNIVERSITY

Week 12

Design with Functions

Learning Objectives

ENGINEERING
TEXAS A&M UNIVERSITY

Functions and Use in Top-Down / Bottom-Up Design

- Create docstrings for functions
- Explain advantages and disadvantages of using a top-down design while utilizing functions
- Utilize functions within a top-down design framework, documenting steps of the process
- Define and give examples of bottom-up design
- Explain and give examples of advantages and disadvantages of bottom-up design
- Utilize a bottom-up design approach as part of creating a Python program
- Compare top-down and bottom-up design
- Define abstraction

Docstrings

Creating help-type information for each of your functions

Docstrings

Docstrings are a special type of comment used to document functions in Python

- What is their purpose?
- What input do they need (parameters)?
- What do they return?

Use the “help” command to access the docstring:

```
help(<function_name>)
```

Ctrl-Q is also a useful shortcut to know. Click in the name of a function, and type Ctrl-Q. You’ll get a pop-up with docstring information and more.

Docstrings

After the function definition, the first line of code should be a string

- Designated with a triple-quote string, which can take multiple lines if needed.

You should **create a docstring for every function***

- What it does,
- Parameters,
- Default parameter values,
- Return values, etc.

*(*Yes, you'll lose points if you don't do so.)*

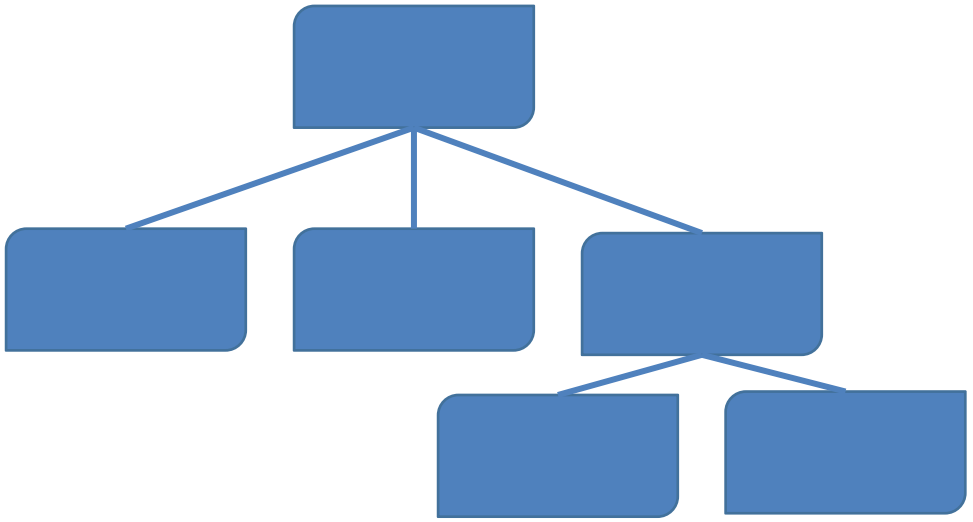
Top-down Design - *with functions*

How do functions work with top-down designs?

Top-Down Design With Functions

Each “block” can be thought of as a function.

- ‘Leaves’ are single functions that don’t call other functions.
- Upper levels of the hierarchy may call many functions.



In the end, each function is easy to understand on its own.

- Can mostly ignore what’s above or below in the hierarchy (except values passed between)

Example

Imagine we have a program to evaluate different models of studying. We want to determine which among several different options will help us learn the most.

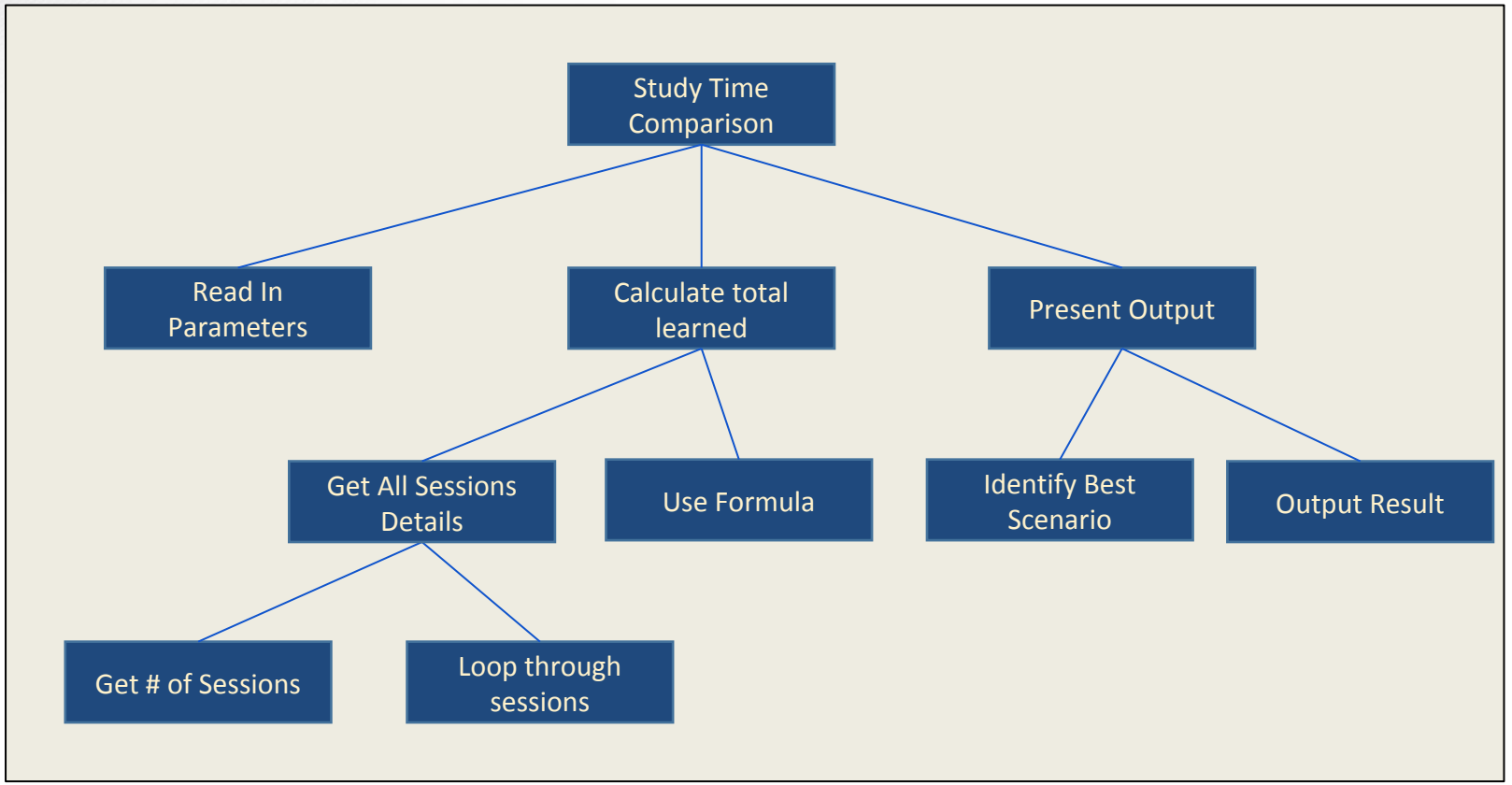
Basic Outline

```
# Get the parameters governing how effective study is
# Loop to get scenarios
# Output results
```

Then break down from there

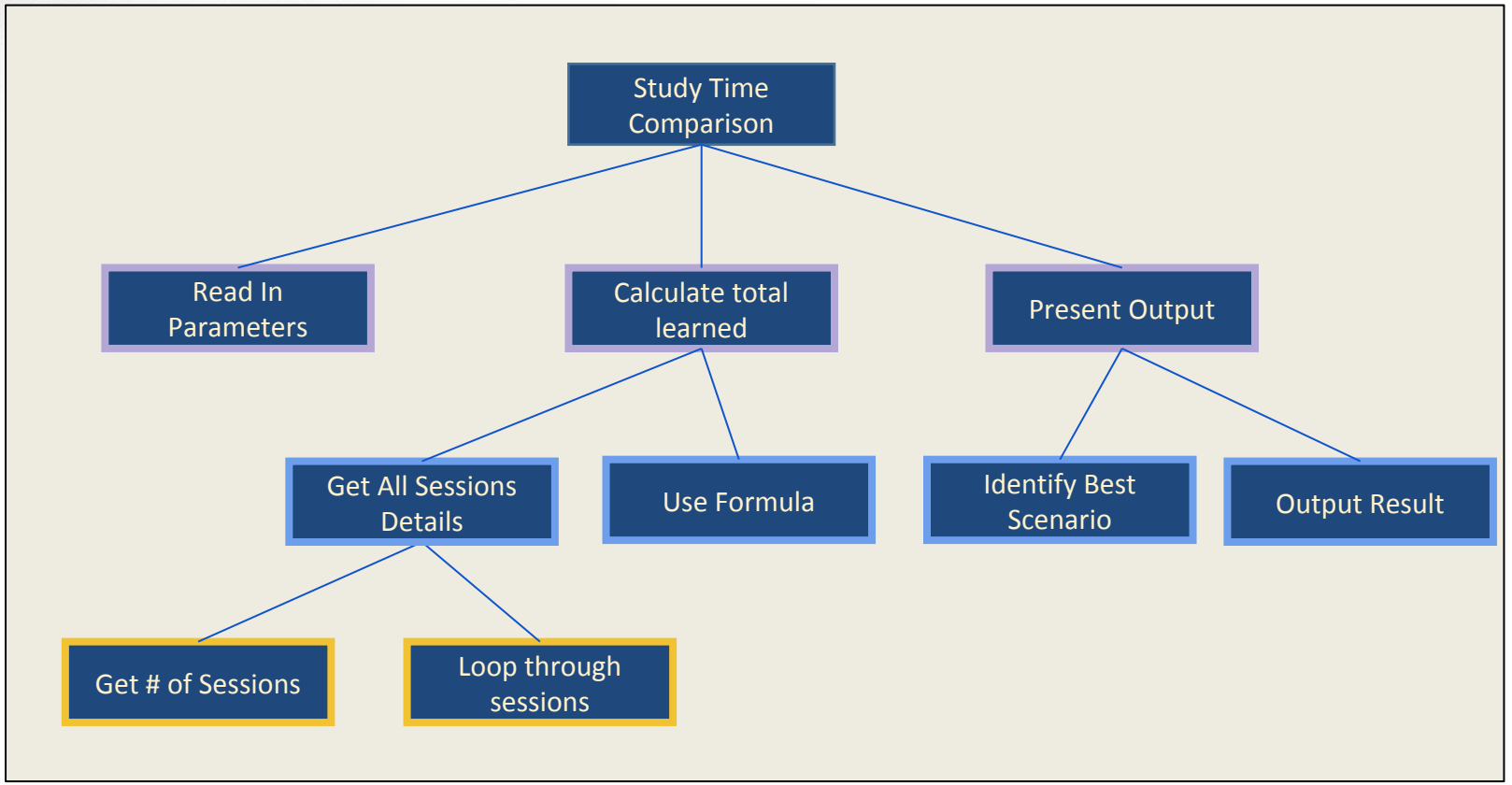
Converting to Code

Make the top-down hierarchy



Converting to Code

Make the top-down hierarchy

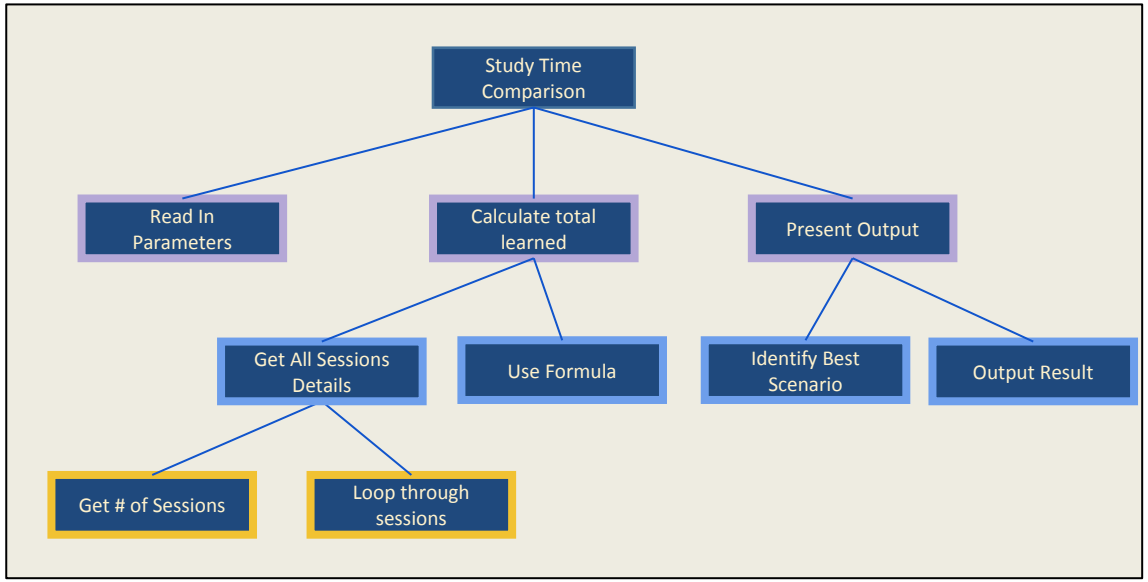



```
def get_num_sessions():
def loop_sessions():
def find_best_scenario():
def output_best_result():
def get_all_sessions():
def use_F():
def readparams():
def calc_learning():
def present_output():
```

Converting to Code

Notice the order of function calls:

- The main program will call first level functions, in order
- Non-leaf functions will call their children functions
- Still no variables or parameters yet!




```
def get_num_sessions():
def loop_sessions():
def find_best_scenario():
def output_best_result():
def get_all_sessions():
def use_F():
def readparams():
def calc_learning():
def present_output():
```

Filling in Code

Next, determine **parameters** to be passed in, and values to be returned.

- For example, “read parameters” does not need any parameters, and returns the information needed to compute study session effectiveness:
  - Starting learning rate
  - Warmup time
  - Steady learning rate
  - Fatigue time
  - Time at which no more learning occurs
- We want to read in this data (e.g. from a user via the console) and return it from the function as a tuple.

Advantages to different functions

Changes are easier!

We have separated **how** the information is provided from the **goal** of the routine!

- The program only needs to know the *what*, not the *how*, of the function.
- The function only needs to know the incoming parameters and the expected return variable/values.

e.g., changing our code to read information from a *file* instead of *user input*, we only change the function

Two options

```
def readparams():
    initrate = float(input("What is the initial rate of concepts learned/minute? "))
    warmuptime = float(input("How long will it take you to warm up? "))
    plateaurate = float(input("What is the rate of concepts learned/minute once warmed up? "))
    fatiguetime = float(input("How long will it take to get fatigued, from the time you start? "))
    stoptime = float(input("What is the point at which you are no longer learning anything? "))
    return (initrate, plateaurate, warmuptime, fatiguetime, stoptime)

(rate_start, rate_steady, t1, t2, t3) = readparams()
```

```
def readparams():
    infile = open("StudyParams.dat",'r')
    initrate = float(infile.readline())
    warmuptime = float(infile.readline())
    plateaurate = float(infile.readline())
    fatiguetime = float(infile.readline())
    stoptime = float(infile.readline())
    return (initrate, plateaurate, warmuptime, fatiguetime, stoptime)

(rate_start, rate_steady, t1, t2, t3) = readparams()
```

Notice the main code is unchanged! It doesn't rely on *how* we get the input.

Top-Down Design with Functions

How far to break down in top-down design?

Think Goldilocks here - 'just right'

- Keep a *clear, single purpose* for a function.
  - If you find one is doing multiple things, split it! (create 'children' or split the node.)
  - Each should be able to be read and understood easily
- Leaves aren't *always* functions
  - if a task needs only 1 line of code, then write one line of main code instead

Bottom-up Design - *with functions*

How does a bottom-up design differ?

Bottom-Up Design with Functions

Create complex code by combining *existing, simpler pieces of code*.

- When planning, think about what you will need to do.
- Write code for basic tasks in short functions.
- As you build more functions, complex code becomes simpler because you have created better tools

  e.g., you created useful functions last week for:

  - calculating a trajectory
  - creating a plot

Bottom-Up Design with Functions

In a true bottom-up design, functions you create will be linked together into subsystems.

- The functions at the bottom have a high chance of reusability.
- Subsystems will link together for more complicated procedures.
- More complicated subsystems are created, until eventually a top-level program is achieved.
- Good intuition is necessary to decide the functions, and the bottom-up design process is often most suitable for a system being built from an existing system.

Bottom-Up Programs

We still have a hierarchy of functions

- Higher level functions call lower level functions
- But, the lower-level functions are designed first

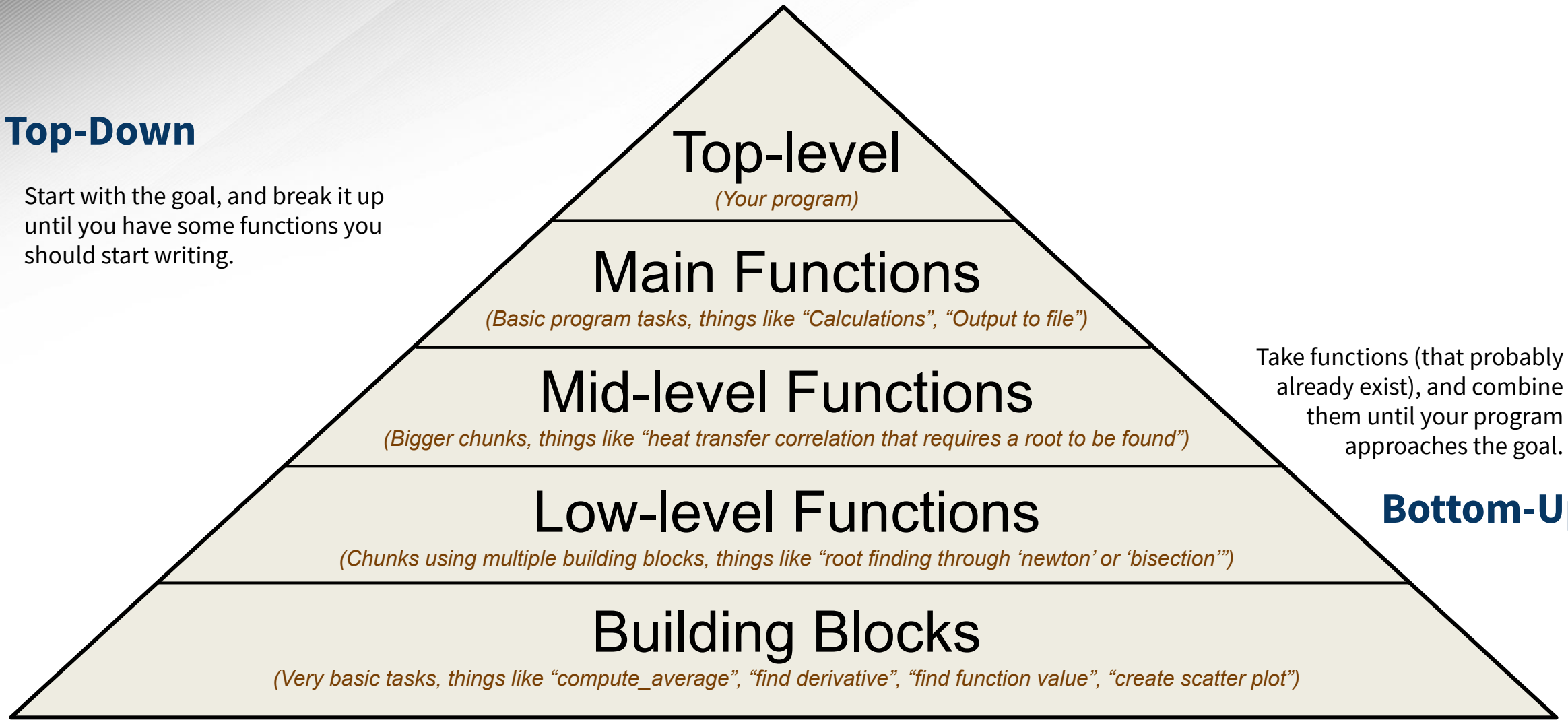
Sometimes top-down and bottom-up will yield similar designs, but not always

Bottom-up approaches tend to be used to develop modules

- More complex routines, built on top of more basic ones
- Can be used in other programs to perform more complex tasks

Top-Down

Start with the goal, and break it up until you have some functions you should start writing.



Take functions (that probably already exist), and combine them until your program approaches the goal.

Bottom-Up

Top-Down vs. Bottom-Up

Top-Down Design

- Creates a hierarchy of functions
- Code is clearly directed toward the goal
- Code can become too specialized, missing chances of re-use
- While developing, much code is in an incomplete state

Bottom-Up Design

- Creates a hierarchy of functions
- Code is often less focused on final goal
- Can promote code re-use
- Code can be tested as each function is finished

Design in Practice

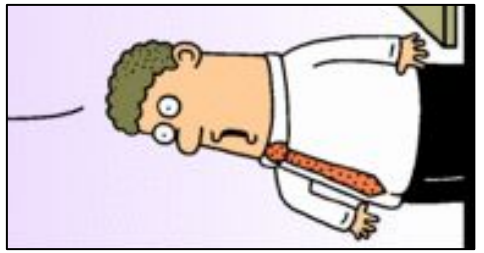
Top-Down and Bottom-Up are not the only design approaches

- Object-oriented design (very common) takes a different approach, combining aspects of several design approaches (look them up!: encapsulation, inheritance, polymorphism)

Programmers seldom use a strict top-down or bottom-up approach, but use aspects of each, depending on the problem / goal.

- Start building bottom-up, thinking about functionality you might need
- Build routines that seem like they'd be commonly called
- Once you have a richer "base" of code, work top-down
- You won't have to go as far in the top-down approach, since you have more useful routines available to you

Abstraction When Designing Functions



A function should not have to worry about the thing calling it

- It just does its job with whatever parameters are passed in

A function should not have to worry about the things it calls

- They should just do their job based on what parameters are passed in

It is very important, especially in larger software projects, to manage complexity.

- Creating and using functions like this is one of the key ways to manage complexity
- There are a variety of other techniques, also, that you'll encounter in more advanced programming