

## Objectives

- Programming with Functions
- Alternative development approaches
- Modules

Feb 4, 2019

Sprenkle - CSCI111

1

## Review

- What are benefits of functions?
- What new development approach did we discuss?
  - What are its steps?

Feb 4, 2019

Sprenkle - CSCI111

2

## Refactoring: Converting Functionality into Functions

1. Identify functionality that should be put into a function
  - What should the function do?
  - What is the function's input?
  - What is the function's output (i.e., what is returned)?
2. Define the function
  - Write comments
3. Test the function programmatically
4. Call the function where appropriate
5. Create a `main` function that contains the “driver” for your program
  - Put at top of program
6. Call `main` at bottom of program

Feb 4, 2019

Sprenkle - CSCI111

3

## Our First Problem

- Create three variables (`i`, `j`, and `result`) to calculate and display  $\text{result} = i^2 + 3j - 5$  for the case where `i=7` and `j=2`. Your code will not look exactly like this formula. Display the result and verify that it is correct. Consider if you were the user of the program and make the program display appropriate output.

So far: We made the function and set up to test it

Feb 4, 2019

Sprenkle - CSCI111

`arithmetic.py`

4

## WHAT MAKES A FUNCTION GOOD?

Feb 4, 2019

Sprenkle - CSCI111

5

### Writing a “Good” Function

- Should be an “intuitive chunk”
  - Doesn’t do too much or too little
  - If does too much, try to break into more functions
- Should be reusable
- Always have comment that tells what the function does

Feb 4, 2019

Sprenkle - CSCI111

6

Development approach:

## BOTTOM-UP DEVELOPMENT

Feb 4, 2019

Sprenkle - CSCI111

7

## Bottom-Up Development

- Define a function
  - Document
- Call the function
  - Test the function
  - Use the function in context

Feb 4, 2019

Sprenkle - CSCI111

8

## Bottom-Up Development Example

- Define a function that
  - Given a team's wins and losses
  - Returns the team's win percentage
- Create a program that
  - Prompts for a team's wins and losses
  - Displays the team's win percentage

`winpercent.py`

## CREATING MODULES

## Where are Functions Defined?

- Functions can go inside of program script
  - Defined before use/called (if no `main()` function)
  - Or, below the `main()` function
- Functions can go inside a separate **module**

Feb 4, 2019

Sprenkle - CSCI111

11

## Benefits of Defining Functions in Separate Module

- Reduces code in primary driver script
- Easier to reuse by importing from a module
- Maintains the “black box”
  - **Abstraction**
- Isolates testing of function
- Write “test driver” scripts to test functions separately from use in script

Feb 4, 2019

Sprenkle - CSCI111

12

## Creating Modules

- Modules group together related functions and constants
- Unlike functions, no special keyword to define a module
  - A module is named by its filename
- Example, `oldmac.py`
  - In Python shell: `import oldmac`
  - Explain what happened

Just a  
Python file!

Feb 4, 2019

Sprenkle - CSCI111

13

## Creating Modules

- So that our program doesn't execute when it is **imported** in a program, at bottom, add

```
if __name__ == '__main__':  
    main()
```

Not important  
how this works;  
just know when to use

- Then, to call **main** function
  - `oldmac.main()`
- Note the files now listed in the directory

Feb 4, 2019

Sprenkle - CSCI111

14

## Creating Modules

- Then, to call **main** function
  - `oldmac.main()`
  - Why would you want to call a module's **main** function?
    - Automation
    - Use **main** function as driver to test functions in module
- To access one of the defined constants
  - `oldmac.EIEIO`

Feb 4, 2019

Sprenkle - CSCI111

15

## Summary: Program Organization

- Larger programs require **functions** to maintain readability
  - Use **main()** and other functions to break up program into *smaller, more manageable* chunks
  - “**Abstract** away” the details
- As before, can still write smaller scripts without any functions
  - Can try out functions using smaller scripts
- Need the **main()** function when using other functions to keep “driver” at top
  - Otherwise, functions need to be defined **before** use

Feb 4, 2019

Sprenkle - CSCI111

16



## Exam Friday

- **Do not panic**
- In-class, on paper
  - Emphasis on critical thinking
- Exam Preparation Document is on course web page
- Similar problems to class and lab
  - Review questions
  - Worksheets
  - Problems
- Content: up through Lab 4
- No broader issue this week

Feb 4, 2019

Sprenkle - CSCI111

17

## This Week

- Lab 4
  - Practicing *functions*
  - Due Friday
- Prelab due before lab tomorrow
- Exam Friday
- No broader issues this week

Feb 4, 2019

Sprenkle - CSCI111

18