Carleton
UNIVERSITY

Canada's Capital University

ECOR1051

Function Design Recipe

Objectives

- Documentation Standards
  - Type Annotations
  - Doc Strings
- Function Design Recipe
  - One example is in the textbook
  - Case study for another example

- Introduction to unit testing, writing unit tests.
- Limitations of using the shell to test functions.
- Reference: Practical Programming, Chapter 3, pp 47-58

Function Headers

- **Previously:**

```
def volume_sphere( radius ):
```

- **Since Python Release 3.5: Type Annotations**

```
def volume_of_sphere(radius: float) -> float:
```

Exercise:

Given a function definition,

```
def volume_of_sphere(radius: float) -> float:
    return 4 / 3 * math.pi * radius ** 3
```

Which of the following will not cause a syntax error?

1. print (volume_of_sphere(5.78)
2. print (volume_of_sphere(5))
3. print (volume_of_sphere("Hello"))

Options

A) 1

B) 2

C) 3

D) 1,2

Documentation String

- **Purpose**
  - Acts as a contract between the function and the caller
  - Conveys to caller the purpose of the function and its limits
    - *In prose*
    - *By examples*

- **Python ignores the docstring - it's for humans to read**

docstring Example

```
def volume_of_sphere(radius: float) -> float:
    """Return the volume of a sphere with the
    specified non-negative radius.

    >>> volume_of_sphere(1)
    4.188790204786391
    >>> volume_of_sphere(1.5)
    14.137166941154067
    >>> volume_of_sphere(0)
    0.0
    """
    return 4 / 3 * math.pi *
```

Prose:

- What the function does (not how)
- Mention each parameter
- [When needed] One line per parameter: Its role and its limits

Examples:

- Sample shell calls with expected return values
- Multiple calls for testing

Sample DocString

```
def fahrenheit_to_celsius(degrees_f: float) -> float:
    """Return the Celsius temperature that is equivalent to
       Fahrenheit temperature degrees_f.
    >>> fahrenheit_to_celsius(32)
    0.0
    >>> fahrenheit_to_celsius(212)
    >>> fahrenheit_to_celsius(-40)
    -40.0
    >>> fahrenheit_to_celsius(80.0)
    26.666666666666668
    """
    return 5 / 9 * (degrees_f - 32)
```

Comment on the "coverage" of set of examples.

help

- Python's built-in `help` function invokes the online help system; e.g.,

  ```
  >>> help(volume_of_sphere)
  ```

  displays text generated from the function's docstring

Function Design Recipe (FDR)

- **Textbook: Designing New Functions: A Recipe**
  - Third Edition, page 47
- A series of steps that guides us, step-by-step, through the process of developing functions
- Goal: understand the function from the caller's perspective before we design and code the body; develop tests early
  - Test-Driven Software Development

FDR – 5 steps

1. **Examples**
2. **Header**
   - Includes type contract
3. **Description**
   - What, not how
4. **Body**
5. **Test**

What steps correspond to parts of the docstring?

Applied Exercise

- **Applying Practical Programming's Function Design Recipe: Chapter 3.11, Exercise 6.**(modified)

- **Following the function design recipe (FDR), define a function that returns the average of three grades, all between 0 and 100 inclusive**

- **See FDR_example**

Applied Exercise: Ambiguity

- *Ambiguity* in a specification can lead a software developer to interpret it in a way that differs from what was intended
- Define a function named convert3 that takes three arguments: the least significant digit of a 3-digit number, followed by the next most significant digit, followed by the most significant digit. The function returns the corresponding number.
- What are the ambiguities?
- Will the FDR help clarify these ambiguities?

Summary

- **DocString:**
  - Function call versus function definition
  - Parameters versus argument
- **FDR**
  - A sequence of steps to follow when writing new functions
  - Uses a testing mindset to first understand what the function will do, from the caller's perspective
    - *Correlates to writing the docstring of the function first.*
  - Developing the body is easy after that!