



CSC I20 Function Design Recipe:

5 steps to design a proper python function

Use these five steps to help you write a correct function in our course

1. Examples
2. Header
3. Description
4. Body
5. Testing



I. Examples

- Pick a name for the function (often a verb or verb phrase). Sometimes a good name is a short answer to the question “What does your function do?”

Let's write a function
that squares any
number and then adds
two to the result

Good name:
`square_plus_two`

- Now, write some examples for the function in the docstring.
 - Choose some standard cases - nothing too tricky

```
"""  
>>> square_plus_two(2.0)  
6.0  
>>> square_plus_two(1.5)  
4.25  
"""
```

2. Header

Let's write a function that squares a number and then adds two to the result

- Write the function header above the docstring.
- Choose a meaningful name for each parameter (often nouns).
- Include the type contract (the types of the **parameters** and **return value**).

Since our function is supposed to square *any* number, it should take any float as input.

`def square_plus_two(n: float) -> float:`

Parameter name

It should also return a float.

3. Description

Let's write a function that squares a number and then adds two to the result

- Before the examples, add a description of what the function does and mention each parameter by name.
- Indent the docstring by four spaces

```
def square_plus_two(n: float) -> float:
```

```
    """
```

```
    Return two more than n squared.
```

```
>>> square_plus_two(2.0)
```

```
6.0
```

```
>>> square_plus_two(1.5)
```

```
4.25
```

```
    """
```

Parameter mentioned
by name



Let's write a function that squares a number and then adds two to the result

4. Body

- Write the body of the function and indent it to match the docstring.

```
def square_plus_two(n: float) -> float:
    """
    Return two more than n squared.

    >>> square_plus_two(2.0)
    6.0
    >>> square_plus_two(1.5)
    4.25
    """

    return n**2 + 2
```

5. Testing

Let's write a function that squares a number and then adds two to the result

- Test your function on all your example cases
- Test other cases, including tricky or 'corner' cases

```
>>> square_plus_two(2.0)
```

```
6.0
```

```
>>> square_plus_two(1.5)
```

```
4.25
```

```
>>> square_plus_two(0.0)
```

```
2.0
```

```
>>> square_plus_two(-1.0)
```

```
3.0
```



Function Design Recipe:

5 steps to design a proper python function

```
def square_plus_two(n: float) -> float:  
    """Return two more than n squared .  
  
>>> square_plus_two(2.0)  
6.0  
>>> square_plus_two(1.5)  
4.25  
"""  
  
    return n**2 + 2
```