Preconditions

- Precondition is a promise
  - If precondition is true, the function works
  - If precondition is false, no guarantees at all
- Get software bugs when
  - Function precondition is not documented properly
  - Function is used in ways that violates precondition

Example: to_centigrade(50.0)

1. Draw a frame for the call
2. Assign the argument value to the parameter (in frame)
3. Execute the function body
   - Look for variables in the frame
   - If not there, look for global variables with that name
4. Erase the frame for the call

```python
def to_centigrade(x):
    return 5*(x-32)/9.0
```

Initial call frame (before exec body)

Call Frames vs. Global Variables

- This does not work:
  ```python
def swap(a,b):
    """Swap vars a & b""
    tmp = a
    a = b
    b = tmp
```

Global Variables
- a 1
- b 2

Call Frame
- a 2
- b 1
- tmp 1

Visualizing Frames: The Python Tutor

Test Cases: Finding Errors

- Bug: Error in a program. (Always expect them!)
- Debugging: Process of finding bugs and removing them.
- Test case: A set of input values, together with the expected output.

Get in the habit of writing test cases for a function from the function’s specification —even before writing the function’s body.
Representative Tests

• Cannot test all inputs
  * “Infinite” possibilities
• Limit ourselves to tests that are representative
  * Each test is a significantly different input
  * Every possible input is similar to one chosen
• An art, not a science
  * If easy, never have bugs
  * Learn with much practice

Representative Tests for \texttt{number_vowels(w)}

• Word with just one vowel
  * For each possible vowel!
• Word with multiple vowels
  * Of the same vowel
  * Of different vowels
• Word with only vowels
• Word with no vowels

Running Example

• The following function has a bug:
  ```python
def last_name_first(n):
    """Returns: copy of \texttt{n} but in the form \texttt{<last-name>, <first-name>}
    Precondition: \texttt{n} is in the form \texttt{<first-name> <last-name>}
    with one or more blanks between the two names"
    end_first = n.find("")
    first = n[:end_first]
    last = n[end_first+1:]
    return last + ', ' + first
```

Looking at precondition when choosing tests:

• Representative Tests:
  * \texttt{last_name_first(Walker White')}
  * \texttt{last_name_first(Walker White')}

Unit Test: A Special Kind of Module

• A unit test is a module that tests another module
  * It imports the other module (so it can access it)
  * It imports the \texttt{cornelltest} module (for testing)
  * It defines one or more test procedures
    * Evaluate the function(s) on the test cases
    * Compare the result to the expected value
  * It has special code that calls the test procedures
  * The test procedures use the \texttt{cornelltest} function

```
def assert_equals(expected, received):
    """Quit program if expected and received differ"""
```

Modules vs. Scripts

Module

• Provides functions, constants
  * Example: \texttt{temperature.py}
• Imports it into Python
  * In interactive shell…
  * or other module
• All code is either
  * In a function definition, or
  * A variable assignment
  * Commands outside functions
  * Does each one in order

```
# Modules/Scripts in this Course
```

# temperature.py
# Functions
def to_centigrade(x):
    """Returns: x converted to C""
...
# Constants
FREEZING_C = 0.0
# temp. water freezes...
# Application code
if __name__ == '__main__':
    print 'Provide a temp. in Fahrenheit: '
    f = float(input())
    c = round(to_centigrade(f),2)
    print 'The temperature is ' + `c` + ' C'

Testing \texttt{last_name_first(n)}

• The following function has a bug:
  ```python
def test_last_name_first():
    """Test procedure for \texttt{last_name_first(n)}"
    cornelltest.assert_equals('White, Walker',
      last_name_first('Walker White'))
    cornelltest.assert_equals('White, Walker',
      last_name_first('Walker White'))
```

Expressions inside of () can be split over several lines.

```
# Application code
if __name__ == '__main__':
    test_last_name_first()
    print 'Module name is working correctly'
```

Message will print out only if no errors.