§10.0-10.7, Py tut §9.0-9.2: Namespaces and Scope

- *Quiz Py10-14 today*
- *Paper Topic (email)*

13 Nov 2006
CMPT14x
Dr. Sean Ho
Trinity Western University

TRINITY WESTERN UNIVERSITY

Review last time (Py tut 8)

- Exceptions:
  - Handling
  - Raising
  - else
  - finally
  - User-defined exceptions
  - Passing auxiliary data with an exception

Quiz09

- Define/describe the following OO terms:
  - Object, class, instance, attribute, method, constructor, overloading
- Create a Python dictionary for the following inventory of apples:

  6 Red Delicious, 12 Fuji, 7 Gala.

  - Write a Python function to print out the inventory in a user-friendly format.
    - Include docstring

Quiz09 answers: #1

- Define/describe the following OO terms: [14]
  - Object: any entity or block of data in an OO language
  - Class: a user-defined container type; collection of variable attributes and method attributes
  - Instance: an object of a given class; variables are to types as instances are to classes
  - Attribute: a variable or method that is a property of an instance or class
  - Method: a function defined within a class
  - Constructor: method which creates new instances of a class
  - Overloading: Functions/operators performing differently depending on the types of the operands

Quiz09 answers: #2 [6]

```
myBag = {'Red Delicious' : 6, 'Fuji' : 12, 'Gala' : 7}

def print_apples(bag):
    """Print what apples we have in the bag."""
    for apple in bag.keys():
        print 'You have %d %s apples.' % \
            (bag[apple], apple)

print_apples(myBag)
```

Namespaces

- A namespace is a mapping from names (identifiers) to objects
  - math.pi is a mapping from the name 'pi' to the float object 3.1415926535...
  - math.pi is in the namespace provided by the math standard library module
- At a given point in the execution of a program, any number of namespaces may be current:
  - Defines what names are valid at that point

Creating namespaces

- The default namespace is present as long as the Python interpreter/compiler is active
  - Contains built-in names like abs(), float(), ZeroDivisionError, etc.
- Each module has a global namespace visible everywhere in that module
  - Variables defined in the outermost level of your Python file
- Each function invocation and class definition also defines a new local namespace
  - Can be nested

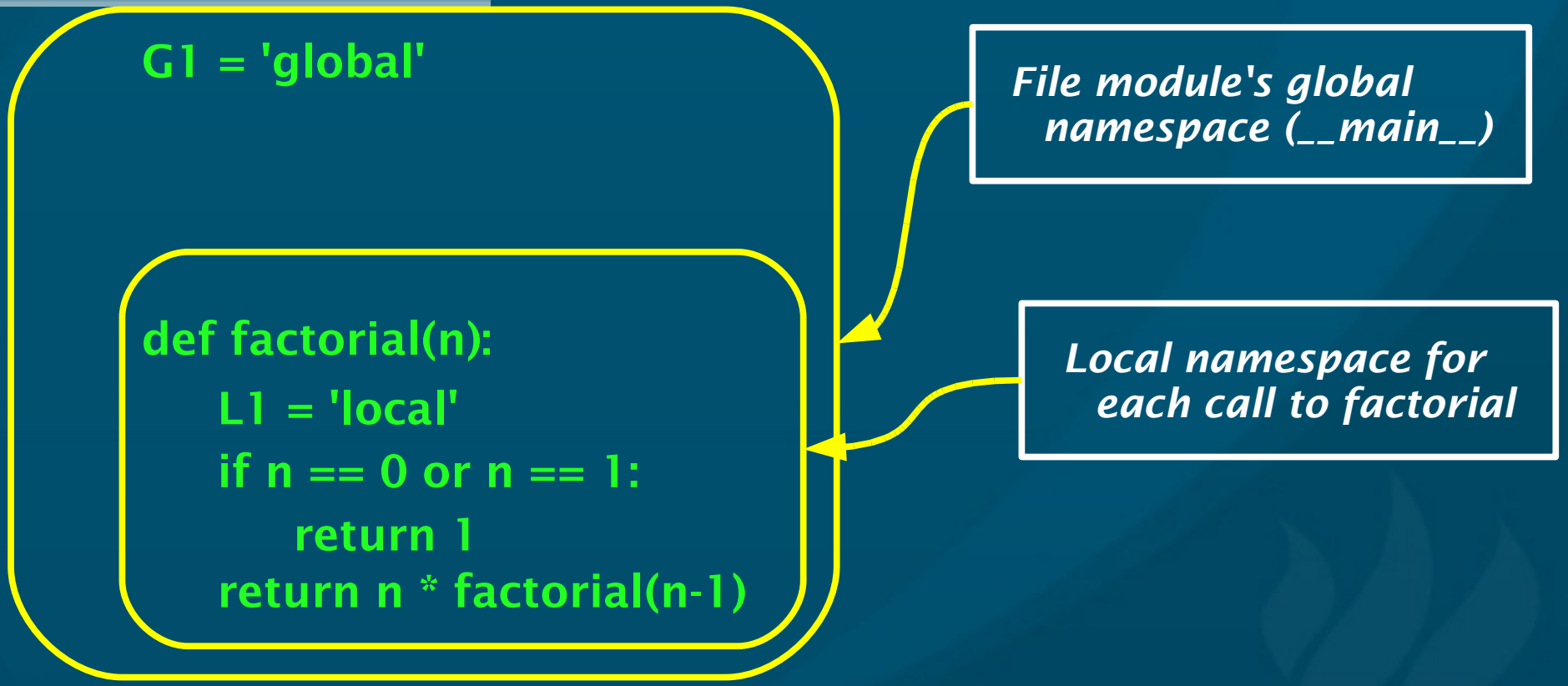
Namespaces avoid name collision

- The point of namespaces is to avoid name collision:
- Names defined in one namespace do not conflict with names defined in another namespace

```
import math
print math.pi       # namespace of math module
pi = 3              # namespace of current file: __main__
```

- Two libraries, or two classes, can define functions with the same name without conflict
  - complex.add() and Fraction.add()

Example of namespaces



Scope

- "A scope is a textual region of a Python program where a namespace is directly accessible."
  - Can access without using module name
    - e.g., pi rather than math.pi
- Scope deals with the order in which namespaces are searched to resolve a name
  - First search local scope
  - Then search enclosing functions/classes
  - Then search global scope for that file/module
  - Then search built-in names

New names add to local scope

- New names are created by:
  - Assignment: x = 5
  - Function definitions: def factorial(n):
  - Class definitions: class Fraction:
  - Imports: from math import *
- New names always add to the local scope

```
def distance(x1, y1, x2, y2):
    from math import sqrt
    return sqrt((x2-x1)**2 + (y2-y1)**2)
sqrt                    # not defined here!
```

The *global* directive

- Names outside the local scope are read-only
  - Attempts to modify them result in creating a new local copy

```
G1 = 'global'
def fun():
    G1 = 'local'     # creates local copy of G1
fun()
G1                   # G1 is unchanged
```

- The global directive says that references to those names refer to the file/module's global scope

TODO

- Lab08 due this week:
  - Robust user input
- HW09 due Wed:
  - Wrapper for open()
- Midterm next week: Wed 22Nov
  - M2 chs9-10
  - Py ch10-14