CSC148H Week 9

Ilir Dema, Michael Miljanovic

Summer 2021

Programs as Data

- An *expression tree* is a structured way of modeling Python code
- By modeling programs as data, we can start thinking about writing programs that operate on other programs
  - A Python interpreter is a program that runs Python code
  - A Java compiler is a program that turns Java code into a sequence of "primitive instructions"
  - PyCharm and PythonTA are programs that analyse Python code and report potential problems

From Expressions to Statements

- An *expression* is a unit of code that, when evaluated, produces a single value
- A *statement* is more general: evaluating a statement can produce a value, and/or has some other effect.
- Every expression is a statement, but not vice versa!

Examples of statements

1. x = 5
2. return 10
3. break
4. if x > 5:
   y = 10
   else:
   y = 15
5. for i in range(10):
   print(i)

Variable bindings

How do we model variables in an abstract syntax tree?
A variable name: the Name class

```
class Name(Expr):
    """A variable name.

    === Attributes ===
    id: The variable name.  """
    id: str
```

- e.g., Name('x'), Name('y'), Name('student_name'), etc.
- But how do we evaluate it?

Mapping variables to values

A *variable environment* is a map from variable names to values. We'll implement this using a Python dict:

{'x': 1, 'y': True}

Name('x').evaluate({'x': 10})

Passing in the environment

```
class Statement:
    def evaluate(self, env: Dict[str, Any]) -> Any:  """Return
        the *value* of this expression,  in the given
        environment.
        """
```

Example

```
>>> expr = Name('x')
>>> expr.evaluate({'x': 10})
10
```

Creating bindings: the Assign class

```
class Assign(Statement):
    """An assignment statement. <target> = <value>

    === Attributes ===  target:
    the variable name  value: the
    expression
    """
```

- e.g., x = 42 + 148

Evaluating an Assign mutates the env

```
>>> stmt = Assign('x', Num(10))
>>> env = {}
>>> stmt.evaluate(env)
>>> env
{'x': 10}
```

Summary

- Name.evaluate: look up the variable name in the current environment
- Assign.evaluate: add a new variable binding to the current environment

Worksheet 1

- The Variable Environment

Worksheet 2

- Control Flow