CSC Lecture 13: Variable Reassignment and Object Mutation

Hisbaan Noorani

October 13, 2020

Contents

1 Ex 1: Reassignmnet and mutation practice

1. .

   (a) Consider this code:

```
1  x = 4
2  y = 5
3  x = 2
```

   Complete the value-based memory model table to show the values of the variables after this code executes. Show both the old and new values of any variables that are reassigned.

Variable	Value
x	~~4~~ $\rightarrow$ 2
y	5

   (b) Consider this code:

```
1  x = 'hi'
2  y = x + 'bye'
3  x = y + x
```

   Complete the value-based memory model table to show the values of the variables after this code executes. Show both the old and new values of any variables that are reassigned.

Variable	Value
x	~~'hi'~~ $\rightarrow$ 'hibyehi'
y	'hibye'

2. Suppose we execute this statement in the Python console.

```
1  numbers = [1, 0]
```

   All of the following statements cause `numbers` to refer to the list value `[1, 0, 8]`. For each one, state whether the statement mutates the original list or reassigns `numbers` to a new list object.

   (a) `list.append(numbers, 8)`
   Mutates

(b) `number = numbers + [8]`
Reassigns

(c) `list.insert(numbers, 2, 8)`
Mutates

(d) `numbers = [numbers[0], numbers [1], 8]`
Reassigns

3. Suppose we execute the following code:

```
lst1 = [1, 0, 8]
lst2 = list.sort(lst1)
```

After the code above is executed, which of the following expressions evaluate to `True`? Circle those expression(s).

(lst1 == [1, 0, 8])	lst1 == [0, 1, 8]
lst2 == [1, 0, 8]	lst2 == [0, 1, 8]

4. Circle the `set` operations that mutate the input set. Try calling `help` on each function, and/or looking them up in A.2 Python Built-In Data Types Reference.

set.intersection	(set.remove)
(set.add)	set.union

5. Suppose we execute the following code:

```
animals = {'fish': {'swim'},
           'kangaroo': {'hop'},
           'frog': {'swim', 'hop'}}
```

Indicate whether each statement will cause an error and, if not, whether the statement will increase the number of key/value pairs in the dictionary:

Statement	Error? (Y/N)	Increases `len(animals)`? (Y/N)
animals['human'] = {'swim', 'run', 'walk'}	N	Y
set.add(animals['monkey'], 'swing')	Y	N
set.add(animals['kangaroo'], 'airplane')	N	N
animals['frog'] = {'tapdance'}	N	N
animals['dolphin'] = animals['fish']	N	Y

6. Read the following function's header and description, and then complete its doctests and implement the function body.

```
def move_item(items: list, other_items: set) -> None:
    """Remove the first item from items and add it to other_items.

    Preconditions:
        - items != []


    >>> numbers_list = [1, 2, 3]
```

```
    >>> numbers_set = {10, 20}
    >>> move_item(numbers_list, numbers_set)
    >>> numbers_list
    [2, 3]
    >>> numbers_set == {10, 20, 1}
    True
    """
    other_items.add(items.pop(0))
```

2 Ex 2: Loopswith collection accumulators

Recall our marriage license dataset, where we represent each row of data using the following data class:

```
@dataclass
class MarriageData:
    """A record of the number of marriage licenses issued in a civic centre
    in a given month.

    Instance Attributes:
      - id: a unique identifier for the record
      - civic_centre: the name of the civic centre
      - num_licenses: the number of licenses issued
      - month: the month these licenses were issued

    Representation Invariant omitted.
    """
    id: int
    civic_centre: str
    num_licenses: int
    month: datetime.date
```

Implement each of the following functions using a loop with an accumulator of the appropriate collection data type. Use mutating operations to avoid creating multiple collection objects.

1. .

```
def filter_by_name(data: List[MarriageData], name: str) -> List[MarriageData]:
    """Return all rows in data with the matching civic centre <name>.

    Equivalent to:
      [row for row in data if row.civic_centre == name]
    """
    rows_so_far = []

    for row in data:
        if row.civic_centre == name:
            rows_so_far.append(row)

    return rows_so_far
```

2. .

```python
def num_issued_by(data: List[MarriageData], centre: str) -> Set[int]:
    """Return the unique numbers of marriage licenses issued in a month at the
    given civic centre.

    Equivalent to:
      {row.num_licenses for row in data if row.civic_centre == name}
    """
    num_so_far = set()

    for row in data:
        if row.civic_centre == name:
            num_so_far.add(row.num_licenses)

    return num_so_far
```

3. .

```python
def marriages_by_centre(data: List[MarriageData],
                        month: datetime.date) -> Dict[str, int]:
    """Return mapping from civic centre name to the number of marriage licenses
    issued by that centre in the given month.

    Preconditions:
        - Each civic centre has only one row of MarriageData for the given month.

    Equivalent to:
      {row.civic_centre: row.num_licenses for row in data if row.month == month}
    """
    record_so_far = {}

    for row in data:
        if row.month == month:
            record_so_far[row.centre] = (row.num_licenses)

    return record_so_far
```