CSCA48 Winter 2016

Week 3 – UML Diagrams

Bo(Kenny) Zhao

University of Toronto Scarborough

January 20, 2016

UNIVERSITY OF TORONTO SCARBOROUGH

Who am I?

- Bo(Kenny) Zhao
- ~~bo.zhao@utsc.utoronto.ca~~
- bo.zhao@mail.utoronto.ca
- or PM through Piazza
- Tutorial: TUT0013 Wednesday 10:00 – 11:00 MW160
- Practical: PRA004 Wednesday 16:00 – 17:00 BV469

Office Hour

LEARNING OBJECTIVES

- At the end of the tutorial, you will be able to …
  - Read UML diagrams
  - Draw UML diagrams

UML

- In A08

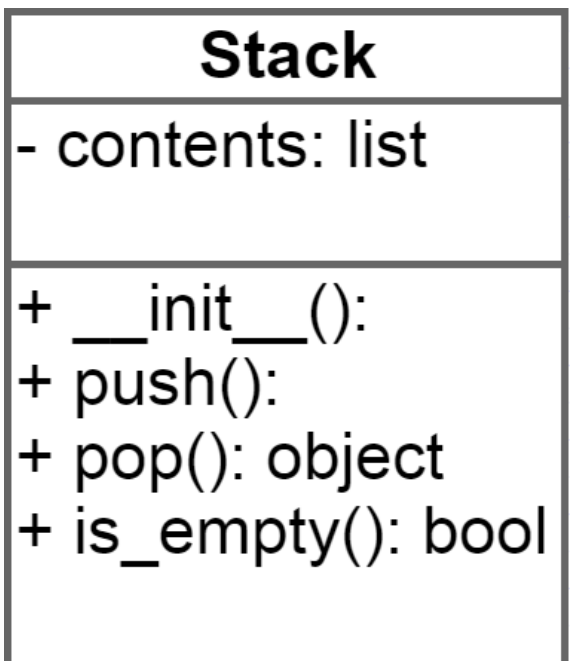
  real-world problem -> code

- In A48

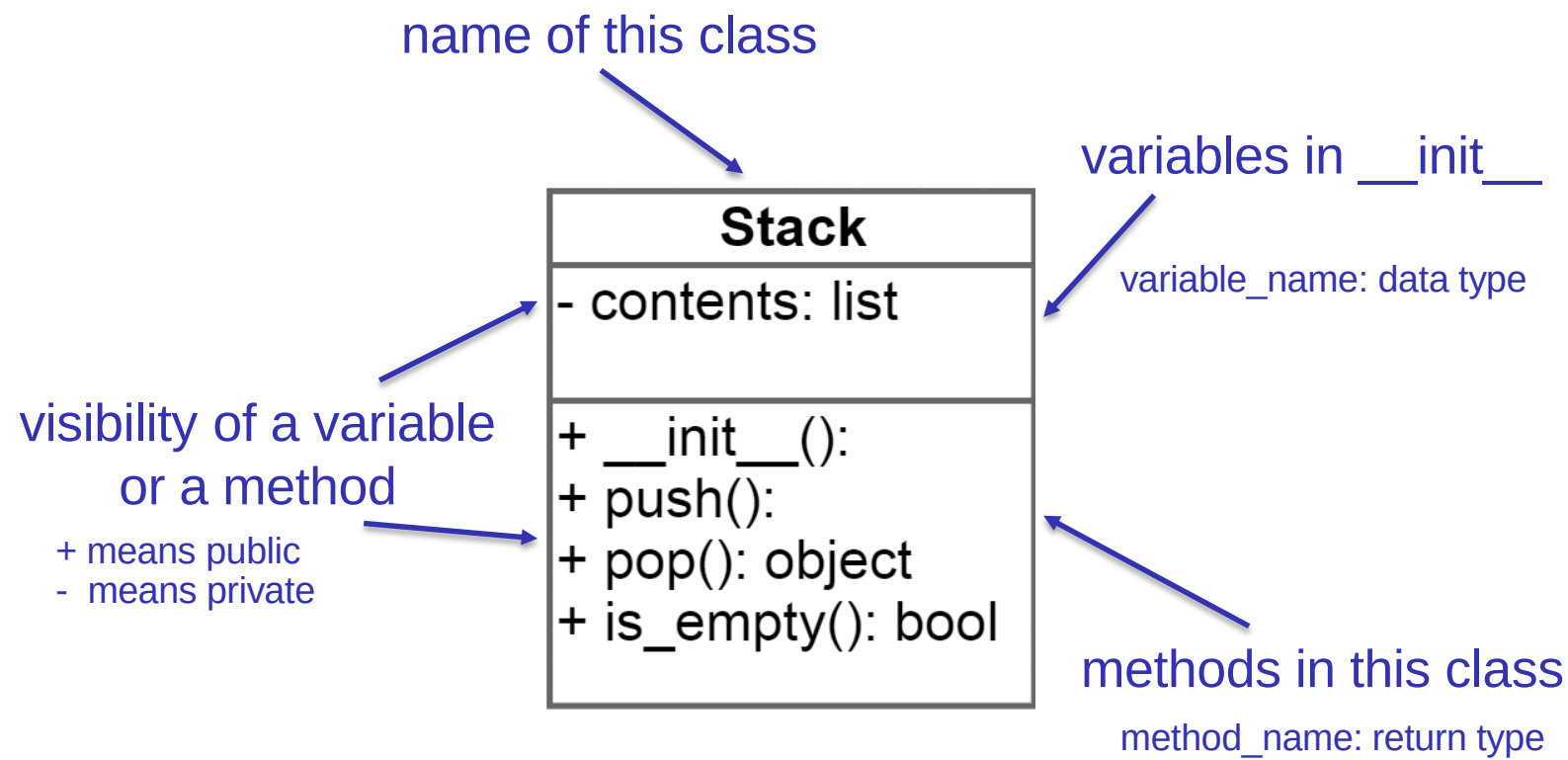
  real-world problem -> design (using UML) -> code

Example

```
1  class Stack:
2      '''A last-in, first-out (LIFO) stack of items'''
3
4      def __init__(self):
5          '''(Stack) -> NoneType
6          Create a new, empty stack.
7          '''
8          self._contents = []
9
10     def push(self, new_obj):
11         '''(Stack, object) -> NoneType
12         Place new_obj on top of this stack.
13         '''
14         self._contents.append(new_obj)
15
16     def pop(self):
17         '''(Stack) -> object
18         Remove and return the top item in this stack.
19         '''
20         return self._contents.pop()
21
22     def is_empty(self):
23         '''(Stack) -> bool
24         Return True iff this stack is empty
25         '''
26         return self._contents == []
```

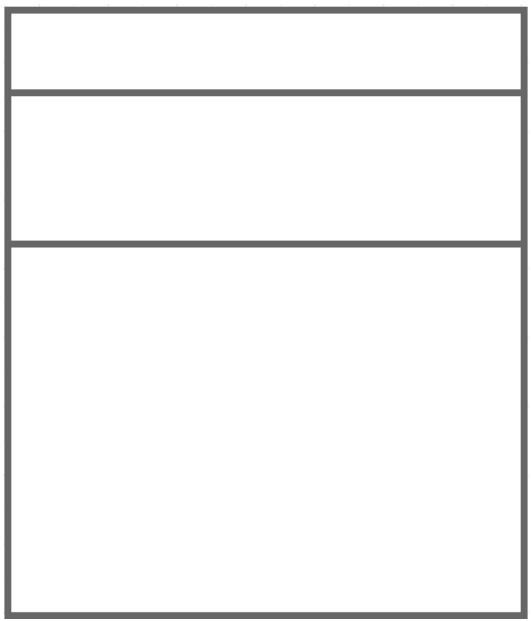


Components of a UML Diagram



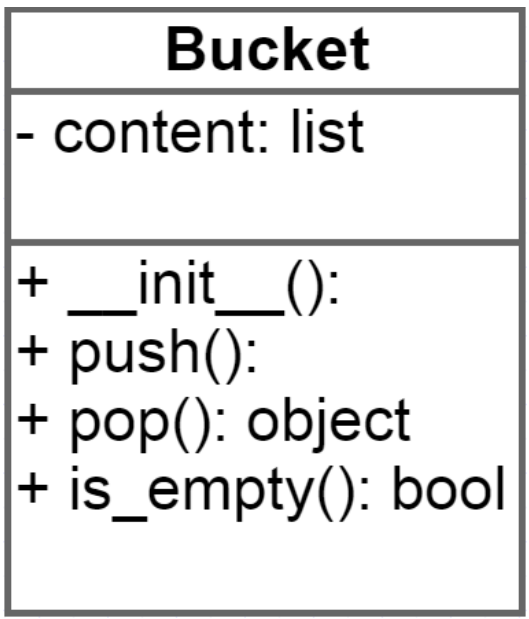
Exercise

```
1  class Bucket:
2      ''' a class representing a bucket(only holds one object) '''
3
4      def __init__(self):
5          ''' (Bucket) -> NoneType
6          Initialize a new empty bucket.
7          '''
8          self._content = []
9
10     def push(self, new_obj):
11         ''' (Bucket, object) -> NoneType
12         Place new_obj into this bucket.
13         REQ: this bucket is not full.
14         '''
15         self._content = [new_obj]
16
17     def pop(self):
18         ''' (Bucket) -> object
19         Remove the object from this bucket.
20         REQ: this bucket is not empty.
21         '''
22         return self._content.pop()
23
24     def is_empty(self):
25         ''' (Bucket) -> bool
26         Return True iff this stack is empty
27         '''
28         return self._content == []
```

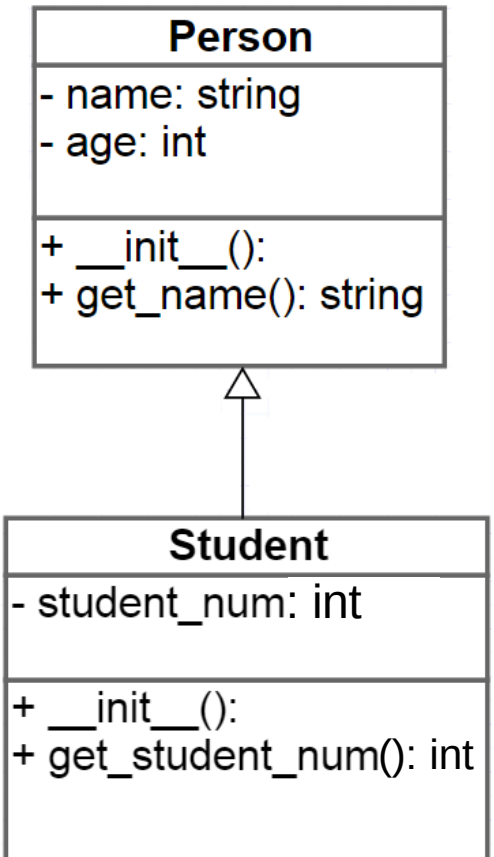


Exercise

```
1  class Bucket:
2      ''' a class representing a bucket(only holds one object) '''
3
4      def __init__(self):
5          ''' (Bucket) -> NoneType
6          Initialize a new empty bucket.
7          '''
8          self._content = []
9
10     def push(self, new_obj):
11         ''' (Bucket, object) -> NoneType
12         Place new_obj into this bucket.
13         REQ: this bucket is not full.
14         '''
15         self._content = [new_obj]
16
17     def pop(self):
18         ''' (Bucket) -> object
19         Remove the object from this bucket.
20         REQ: this bucket is not empty.
21         '''
22         return self._content.pop()
23
24     def is_empty(self):
25         ''' (Bucket) -> bool
26         Return True iff this stack is empty
27         '''
28         return self._content == []
```

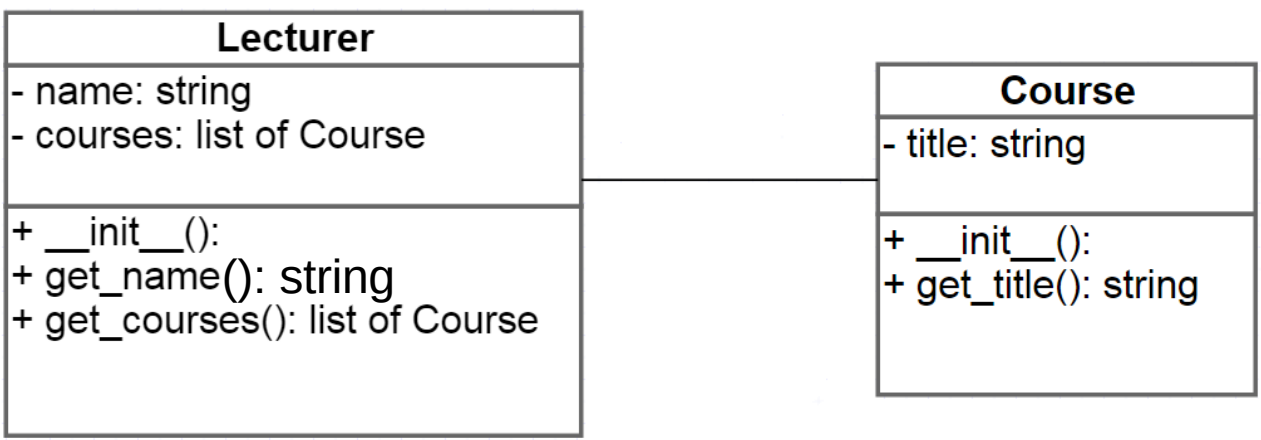


Inheritance



- Do not redraw variables and methods inherited from parent class.
- Assume everything in parent class is inherited.

Association



Exercise

- We need to represent books and authors.
- Author should have a first and last name, a mailing address and all books that belong to this author. Authors are either salaried or contractors. We should be able to see for any author what are the books he/she wrote, and how much they've been paid.
- Book should have a title, number of pages and all authors that contribute to this book. Sometimes we need to add extra pages, so it would be good to have a simple way of doing that. Now, given a book, I should be able to get the information and all of its authors.
- Books can be paperback or hardcover, and hardcover books can either be regular or special edition. Each type has its cost. Paperbacks cost less, and a special edition's price depends on whether or not it has a certificate with it. The number of copies we print will also depend on the type of book.For paperbacks, the total copies will based on a formula; for hardcover it depends on the author; and for special editions, we always print a fixed number.

Exercise(looking for nouns)

- We need to represent books and authors.
- Author should have a first and last name, a mailing address and all books that belong to this author. Authors are either salaried or contractors. We should be able to see for any author what are the books he/she wrote, and how much they've been paid.
- Book should have a title, number of pages and all authors that contribute to this book. Sometimes we need to add extra pages, so it would be good to have a simple way of doing that. Now, given a book, I should be able to get the information and all of its authors.
- Books can be paperback or hardcover, and hardcover books can either be regular or special edition. Each type has its cost. Paperbacks cost less, and a special edition's price depends on whether or not it has a certificate with it. The number of copies we print will also depend on the type of book.For paperbacks, the total copies will based on a formula; for hardcover it depends on the author; and for special editions, we always print a fixed number.

Exercise(looking for nouns and verbs)

- We need to represent books and authors.
- Author should have a first and last name, a mailing address and all books that belong to this author. Authors are either salaried or contractors. We should be able to see for any author what are the books he/she wrote, and how much they've been paid.
- Book should have a title, number of pages and all authors that contribute to this book. Sometimes we need to add extra pages, so it would be good to have a simple way of doing that. Now, given a book, I should be able to get the information and all of its authors.
- Books can be paperback or hardcover, and hardcover books can either be regular or special edition. Each type has its cost. Paperbacks cost less, and a special edition's price depends on whether or not it has a certificate with it. The number of copies we print will also depend on the type of book.For paperbacks, the total copies will based on a formula; for hardcover it depends on the author; and for special editions, we always print a fixed number.

More Practice

- Course webpage(search CSCA48)

  -> Practicals Anti-Lectures Office Hours
  -> Practical Questions
  -> Week 3
  -> questions h) to m)

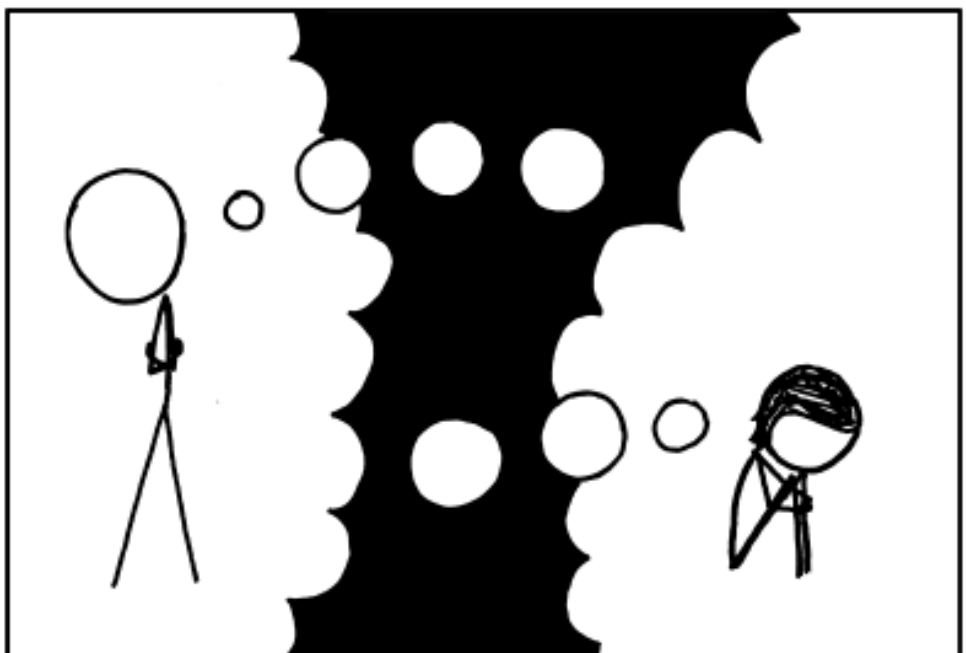
http://www.utsc.utoronto.ca/~bharrington/csca48/practicals/week3.pdf

Summary

- Don’t focus on the details
  - string and str are both acceptable
  - a method without return type means it returns None

- Encapsulation
  - Don’t make a variable public unless you have a good reason

Quiz



https://xkcd.com/817/