Week 4: COMP-801 - Integrated Computing Practice

Agenda

<<<<<<< HEAD

- Feedback to **Active Reading 3**
- Introduction to classes
- More on functions
- More on iteration
- Interleave lab work with understanding of concepts, techniques, and tools
- Review: Variable scope
- Due next week
- Exam next week

Active Reading Feedback and Discussion

- Feedback is summarized in the AR reports posted in Discord
- Let's select and discuss
  - 2+ questions from the functions section
  - 1+ question from the iteration section

  =======
- Feedback to **Active Reading 3**
- Introduction to classes
- More on functions
- More on iteration
- Review: Variable scope
- Due next week
- Exam next week

Active Reading Feedback and Discussion

- Feedback is summarized in the AR reports posted in Canvas
- Let's select and discuss
  - 2+ questions from the functions section
  - 1+ question from the iteration section

“ “ “ “ “ “ “ 424b2ef (remove lab3 from week4 slides) ” ” ” ” ” ” ”

📖 Variable Scope

The **scope of a variable** is the code (set of statements) where the variable is known and can be referenced (or accessed).

```
# define module-level variable in `practice/week4.py`
message = 'Hello'
def greet1():
    print(f'Inside function: {message}')

print(f'Outside function: {message}')
greet1()
```

📖 Functions

- Arguments are passed to a function
- Parameters are **local** in scope
- Print vs. return statements
- Passing mutable objects
- Side effects

📖 Classes - Syntax

How we write a class (syntax)

Class definition statement has two parts:

1. Class header has keyword **class** and class name
2. Class body has **methods**:
    1. **constructor**: `__init__(self, ...)`
        1. Creates an object of the class
        2. Defines **instance variable**
        3. Constructor's parameters initialize the **instance variable**
    2. **instance methods** define behaviors that objects of the class have
    3. Other types of methods (we'll learn about later)

📖 Constructor Method

- `Person` class has a constructor that defines **instance_varialbes**:
  - first name, last name, and age
  - There are attributes of any object of type `Person`

```
class Person
    def __init__(self, first_name, last_name, age)
      self.first_name = first_name
      self.last_name = last_name
      self.age = age
```

- The constructor initializes the attributes with values from its parameters
- `self` parameter is the place holder for the address of the object constructed from this class

📖 Classes - Semantics and Use

What a class means (semantics) Encapsulates

- attributes or properties: e.g., **instance variables**, that all objects have
- behavior: e.g., **instace_methods** that objects perform

How we use a class (utility)

- Create objects of the class by calling the class constructor

Exam 1 format

- In-person, pencil-and-paper written examination
- To answer questions and solve problems
- Expected duration: 15 to 30 minutes
- Midterm exercises and problems are identical or similar to textbook exercises and problems
- Bring pencils and eraser

Exam 1 Resources

- Active reading progress reports
- Your work in labs and homework assignments
- Feedback to labs and homework
- Discord communication
- Q&A with CAs and instructors

Exam 1 Concepts include

- Value, data type, operator, expression
- Statement: simple, composite
- Function definition, function call
- Class definition, attributes, methods, constructor
- Built-in data types: `str`, `list`
- Accumulation pattern